

Bayesian Expert Systems and Multi-Agent Modeling for Learner-Centric Web-Based Education

by

Ralph Rizkallah Rabbat

Master of Science, Massachusetts Institute of Technology, 2002

Bachelor of Engineering in Computer and Communication Engineering,
American University of Beirut, 2000

Submitted to the Department of Civil and Environmental Engineering in partial
fulfillment of the requirements for the degree of

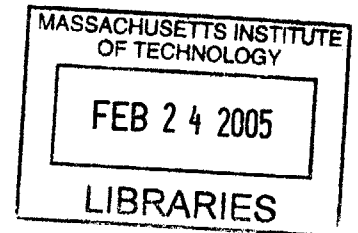
BARKER

Doctor of Philosophy in Information Technology

at the

Massachusetts Institute of Technology

February 2005



© 2005 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: _____

Department of Civil and Environmental Engineering
January 10, 2005

Certified by: _____

Steven R. Lerman
Professor of Civil and Environmental Engineering
Thesis Supervisor

A handwritten signature in black ink, appearing to read "S. Lerman".

Accepted by _____

Andrew J. Whittle
Chairman, Departmental Committee on Graduate Studies

Bayesian Expert Systems and Multi-Agent Modeling for Learner-Centric Web-Based Education

by

Ralph Rizkallah Rabbat

Submitted to the Department of Civil and Environmental Engineering on January 11, 2005, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of Information Technology

Abstract

Online distance education provides students with a wealth of information. When students submit course-related search term queries, the search engine returns the search hits based on keyword and topic match. A student's particular learning style is not taken into consideration. For instance, a visually oriented student may benefit more than others from viewing videos and interacting with simulations.

We address this problem by designing and developing a knowledge-based system for the initial assessment of students' learning styles. Each student's membership in a learning style dimension (e.g. visual or verbal) is estimated probabilistically. We reach this probability value by using a sequential Bayesian approach to administer a dynamic questionnaire that aims to attain a desired confidence level estimate with the minimal number of questions.

A multi-agent online tutoring system uses this initial learning style model to start suggesting learning material matching the student's style. Each agent is an expert in a learning style dimension and can suggest the learning materials matching the student's style. In addition, these agents closely follow the student's evolving preferences and continuously update the stochastic model based on the student's online activities. When the student searches for course material, the multi-agent system delivers the search matches in a cycle-free preference order influenced by the students' multi-dimensional learning style model.

Thesis Supervisor: Steven Lerman

Title: Professor of Civil and Environmental Engineering

Acknowledgements

I would first and foremost like to thank my advisor Steven Lerman, for his encouragement, constant guidance during the research, and the writing of this thesis. Steve did the best a student could ask from his advisor. He made the years that I spent at MIT enjoyable and a great learning experience. He helped me in both my technical skills as well as my writing and documentation skills.

I would like to thank Chris Dede, Professor in Learning Technologies at Harvard, for his help and advice throughout this thesis work. He helped me identify the key pedagogical directions to explore and therefore contribute to the learning and technological education literature. His experience, resources and advice were key components for the success of this research.

Jerome Connor, professor at MIT, gave me advice and thoughtful insights about the learning challenges in virtual classrooms and how my contributions can fit the current needs of learning in a technological environment.

I would also like to thank Cynthia Stewart, who has helped my PhD's thesis accomplishment tremendously by providing me with advice, reminding me of deadlines, and being of help at both the logistical and academic levels. She was the person I could rely on to help me when needed with departmental and personal issues and conflicts.

My immediate family represents for me the key to my dedication and hard work during my PhD studies. They have always been a constant source of comfort, encouragement and words of wisdom. Mom, Ron and Richard, I thank you from the deepest of my heart for all the great advice and the love that you have given me. My late father, Rizkallah, has always been my source of inspiration and my role model. I would like to thank him for all the great moments that we had together and the best upbringing a

son could ever wish for. I will never forget his dedication and unfettered love, and I will keep making him the proudest of all fathers.

Let me also mention friends, Lebanese, American and international. You made my stay at MIT a memorable phase. I enjoyed your energy, wit, jokes, and the great discussions we have had. I would like to mention Karim, who was a terrific friend, a reliable resource, and a great confidant during my stay in Boston and at MIT. I would also like to thank my lifelong friends Paul, Colette and Fernando for reminding me of long-lasting friendship. I would also like to mention my lab mates and colleagues here and gone, Jean-Claude, Issam, Bassam, Mesrob, Karim Y, Jed and Charuleka.

The Lebanese Club at MIT was a great experience for me. It helped me understand that my point of view was not the correct one all the time. I learnt things about my country that I had never known because I was on “the other side.” The frequent and sometimes heated discussions that I had with Jad Karam and Loai Naamani were eye-opening experiences that gave me a great deal of knowledge. The club has helped me understand the differences in attitudes between the cultures, ages, religions and backgrounds. I put my best efforts to keep the Lebanese students have close bonds, and I managed to give a great positive image of Lebanon to the MIT community.

It has been about five years that I am living in Boston, away from Lebanon, where I spent my younger years. Although I miss it quite a bit, I also feel torn between my allegiance to my new country, America, that has embraced me and of which I have become a citizen, and my hometown Beirut. Beirut and Boston will always represent what I like and dislike most in the world. They carry in their hearts the best and the worst days of my life and for that, I love you both.

**To My Late Father Rizkallah,
Josephine, Ronald and Richard**

Thank You!

Table of Contents

Acknowledgements.....	3
Table of Contents.....	6
Tables.....	10
List of Equations.....	11
 Chapter 1 – Introduction.....	 13
1.1 Overview.....	13
1.2 Problem Definition.....	14
1.3 Motivation.....	15
1.3.1 Online Education	16
1.3.2 Business Perspective.....	17
1.4 Scope of the System.....	18
1.4.1 The PIVoT Project	19
1.5 Goals	19
1.5.1 Educational Goals	20
1.5.2 Computational Goals	21
1.5.2.1 MAS Architecture.....	22
1.5.2.2 Scalability	23
1.5.2.3 Platform Neutrality	23
 Chapter 2 – Literature Review.....	 25
2.1 Adaptive Hypermedia	25
2.1.1 Adaptive navigation	26
2.1.1.1 Direct guidance	26
2.1.1.2 Adaptive ordering	26
2.1.1.3 Hiding	26
2.1.1.4 Adaptive Annotation.....	27
2.1.2 Adaptive Presentation	27
2.2 Learning Styles	28
2.2.1 History of Learning Styles.....	28
2.2.2 Learning Styles Approaches	29
2.2.2.1 Dunn and Dunn Model.....	29
2.2.2.2 Kolb Model	30
2.2.2.3 Myers-Briggs Type Indicator.....	33
2.2.2.4 4MAT Model	34
2.2.2.5 Reichmann and Grasha’s Model.....	35
2.2.2.6 Gardner’s Multiple Intelligences	35
2.2.2.7 Hermann Brain Dominance Instrument.....	36
2.2.2.8 Felder-Silverman Learning Style Model	37
2.2.3 Issues with Learning Style Models.....	39
2.3 Artificial Intelligence in Education.....	40

2.3.1	Expert Systems.....	42
2.3.2	Probabilistic Inference and Bayesian Networks	43
2.3.3	Software Agents.....	44
2.3.3.1	Characteristics and Classification.....	45
2.3.3.2	Collaborative Agents	48
2.3.3.3	Interface Agents	48
2.3.3.4	Pedagogical Agents.....	49
2.4	Intelligent Tutoring Systems.....	49
2.4.1	Learner Modeling.....	51
2.4.1.1	Probabilistic Models	51
2.4.1.2	Non-Probabilistic Models	52
2.4.2	Cognitive and Pedagogical Approaches	52
Chapter 3	– A Knowledge Based System for Learning Style Assessment	55
3.1	Introduction.....	56
3.2	Learning Styles Assessment Tools	58
3.2.1	Soloman and Felder Index of Learning Styles Questionnaire	58
3.2.2	The VARK Learning Style Inventory.....	59
3.2.3	Kolb Learning Style Inventory	59
3.3	Learning Styles Approach.....	61
3.4	Learning Style Dimensions.....	62
3.4.1	Perception Styles.....	63
3.4.2	Reception Styles.....	63
3.4.3	Processing Styles	63
3.4.4	Interaction Styles	64
3.5	Teaching Style Dimensions	65
3.5.1	Content Level.....	66
3.5.2	Presentation Level.....	66
3.5.3	Participation Level.....	67
3.5.4	Interaction Level	67
3.6	Expert Learning Style Assessor	68
3.6.1	Overview	68
3.6.2	Importance of the KBS	68
3.6.3	Attributes of the System.....	69
3.6.4	Nature of the Questions	69
3.6.5	Categorization of the Learning Style Dimensions	70
3.7	Assessment of Learning Style.....	72
3.7.1	Adaptive Questionnaire Requirements	72
3.7.2	Bayesian Updating.....	73
3.7.3	Question Sequence and Value of Information	75
3.7.3.1	Formulation of Information Value.....	75
3.7.3.2	Question Sequence.....	78
3.7.4	Determination of the Learning Styles	80
3.8	Results.....	80
Chapter 4	– Agreement of Multi-Agent System Recommendations	83
4.1	Related Work	84

4.2	Multi-Agent System Functionality	85
4.3	Individual Agents' Ranking of Educational Media	87
4.4	Multi-Agent System and Conflict Resolution.....	90
4.5	Ranking and Voting	92
4.5.1	Cycles.....	95
4.6	Ranking Methods	96
4.6.1	Borda's Method	97
4.6.2	Kohler and Arrow-Raynaud.....	99
4.6.2.1	Kohler's Approach.....	100
4.6.2.2	Arrow-Raynaud's Method	101
4.7	Discussion of the Methods.....	102
4.8	Multimedia Ranking Results.....	103
4.9	Conclusion	105
Chapter 5	– Software Architecture	106
5.1	Intelligent Education and the Web.....	107
5.2	Data Design.....	109
5.2.1	Metadata.....	110
5.2.2	Ontology	111
5.2.3	Content Structure and Identification	112
5.2.4	Media	113
5.2.4.1	Media and Metadata.....	114
5.2.4.2	Property Classification.....	115
5.3	System Architecture.....	116
5.3.1	Technology and Programming.....	118
5.3.2	Server-Database Interaction.....	119
5.3.3	Serialization	120
5.4	Multi-Agent System Architecture.....	121
5.4.1	Expert System Integration.....	122
5.4.2	Agents' Observations of Student Activity	123
5.4.3	Online Student Activity	123
5.4.4	Student Profile and Database	124
5.5	Summary.....	125
Chapter 6	– Conclusions and Recommendations	126
6.1	Contributions.....	126
6.1.1	Technical Contributions.....	127
6.1.1.1	Plug-in system.....	128
6.1.1.2	Conflict Resolution	128
6.1.2	Pedagogical Contributions	129
6.2	Future Research	130
6.2.1	Educational Assessment.....	130
6.2.1.1	Grade Improvement	131
6.2.1.2	Online Time	131
6.2.1.3	Relevance of Search Results.....	131
6.2.2	Expert System Improvement.....	132
6.2.3	Teaching Strategies.....	133

6.2.4	Intelligent Educational Environments.....	133
6.2.5	Corporate Applications	134
6.3	Conclusions.....	135
Appendix A.....		137
References.....		140

Tables

Table 2-1: AI Systems	41
Table 3-1: Learning Styles and Corresponding Observed Criteria.....	64
Table 3-2: Corresponding Teaching-Learning Styles.....	66
Table 3-3: Conditional Likelihood for Q_1	72
Table 3-4: Posterior Probability Matrix for Q_1	76
Table 3-5: Perfect Information Matrix for Q_1	77
Table 3-6: Penalty Matrix for Q_1	77
Table 3-7: Likelihood Matrix $p[A_{ir} C_{jk}]$ For Q_2	78
Table 3-8: Posterior Probability Matrix for Q_2	78
Table 3-9: Perfect Information Matrix for Q_2	79
Table 3-11: Penalty Matrix for Q_2	79
Table 3-13: Results of Learning Style Assessment	81
Table 4-1: Distribution of Conditional Likelihood of Media Over Types.....	89
Table 4-2: Cycles Illustration.....	95
Table 4-3: Voters' Preferences	96
Table 4-4: Agent's Assessment of Subject U's Preferences Using the Borda Count	98
Table 4-5: Outranking Matrix.....	100

List of Equations

$p[C_{jk} A_{ir}] = \frac{p[C_{jk}] \times p[A_{ir} C_{jk}]}{p[A_{ir}]}$	74
$Posterior(r, j) = \frac{p[C_{jk}] \times p[A_{ir} C_{jk}]}{p[A_{ir}]}$	76
$S_r = \{ j Posterior(r, j) = \max_d Posterior(r, d) \}$	76
$\forall r, j \quad Perfect(r, j) = \begin{cases} 1/ S_r ; & j \in S_r \\ 0; & otherwise \end{cases}$	76
$\forall r, j \quad Penalty(r, j) = Posterior(r, j) - Perfect(r, j) $	77
$EVI(Q_i) = \sum_{r,j} Perfect(r, j) - \sum_{r,j} Penalty(r, j)$	78
$p[A] = \sum_n p[A B_n] \times p[B_n]$	89
$p[video] = p[video visual] \times p[visual]$	89
$+ p[video verbal] \times p[verbal]$	
$+ p[video neutral] \times p[neutral]$	

This page is intentionally left blank.

Chapter 1 – Introduction

1.1 Overview

Online learning has become widespread in the past few years, and according to an AOL advisor, the online student population is increasing by thirty percent a year [37]. In addition to individuals earning degrees, businesses have expanded their corporate learning environments to reduce teaching costs, and stay up-to-date with the rapid pace of technological innovation.

Distance learning, also known as e-learning or online education, presents students with a wealth of information. It has become popular due to the power and low cost of Web-based delivery systems, as well as the comfort of learning at home [114]. It can be transmitted through different media and involves various sets of activities. The teaching channels can be either synchronous or asynchronous. The latter being richer in content, format, and diversity of learning activities involved, this research aims at making asynchronous online learning a more customized experience to students' preferences and individual learning styles.

This thesis proposes an educational, architectural, and mathematical approach to providing personalized online teaching material. The application field is that of a physics online supplementary website at the Massachusetts Institute of Technology (MIT), the Physics Interactive Video Tutor (PIVoT) [52]. This research specifically aims at delivering online material to students according to their learning style preference, thereby achieving a more efficient and customized learning experience.

This chapter presents the guiding principles of this thesis. The sections below discuss the problem at hand, the motivation behind the work, and the goals of this research.

1.2 Problem Definition

Students' learning styles or preferred ways of learning vary according to different parameters. Their background and previous experiences, their mental and cognitive predispositions, the learning environment, and even the time of the day when they are studying all play significant roles in affecting students' learning experiences, and their capability to assimilate the course material. The focus of this research is on a one-on-one interaction between the students and the online teaching material. The traditional way of interaction is to have the material laid out for them, and students browsing through it, searching for relevant topics, and learning the material. Course management systems do not usually "know" the student and simply present the material identically to all students without taking into account each student's preferred way of learning. Education and psychology researchers have tried to understand how students learn; this research work will be summarized in Chapter 2. Primarily, these theories were applied to traditional

classroom environments and were sometimes successful in raising students' achievement. Online applications of learning theory are still scarce and have mainly focused on aiding students in problem-solving steps.

Due to the richness of asynchronous education, online students are presented with a variety of learning material to discover. Therefore, the potential of spending a long time trying to search for the right information and the frustration of not finding a delivery method suitable to one's preferred way of learning can both negatively affect students' online learning and alienate them from a potentially beneficial and efficient learning experience.

Our focus is to build a model of each student's preferred learning style (LS) and deliver the online course material that suits the model's predictions. Students are not always aware of their preferred learning methods, but rather go through the various available materials and activities with the hope that a certain delivery medium will teach them the course content, with the medium being visual, verbal, auditory, kinesthetic/active, or interactive.

In addition, students' learning styles need not be constant over time or when the learning environment changes (e.g., in location or setting). There is no research evidence that proves that individuals' LSs are constant over time. Snow [113] and Kozma's [71] findings indicate that learning styles can vary with different tasks or different learning contexts.

1.3 Motivation

The motivation for this research comes from a combination of pedagogical and market needs. On the one hand, there is a need to make online learning experiences more

customized to each student's learning style, thus more effective. On the other hand, companies and educational institutions need to make online learning adopted more widely in order to cut costs or increase the size of their markets. By ensuring a positive and customized online experience, students would learn more efficiently and effectively.

1.3.1 Online Education

Briefly explained, traditional synchronous teaching is delivered by having a teacher lecturing at some location, with the students following the lecture on-site or remotely. Remote students can send questions to the teacher by text (e.g. instant messaging) or voice if they have audio permissions, and the teacher will choose to either answer the questions immediately if they are relevant to most students or to delay them until the end of the lecture. Asynchronous teaching has the potential and capability to deliver richer media: pre-recorded video segments of lectures, problem sets, interactive simulations and animations, diagrams and charts, textbook explanations, frequently asked questions, discussion boards among peers and teachers, e-mail questions and answers, etc.

The major advantages of asynchronous learning are in the variety of teaching media and the time flexibility as discussed previously. Students do not have to be at their computers at a specific time, they are not geographically bound, and they are not constrained by the duration of the learning activity. In addition, if students do not grasp a topic explanation from the first time, they can go over it again. However, an apparent downside of asynchronous learning is the possibility of excessive learning material presented to the student. The student may be inclined to go through all the material

relevant to a topic in order to be competent, even though this may not be needed and is likely inefficient.

When information about a topic is presented in different formats and involves various activities, different students are interested in diverse ways of learning due to their individual learning styles. For instance, visually oriented students may benefit more than others from viewing videos and interacting with simulations. In order to make the learning media customized to a student's interests and preferred way of learning, one way to approach this problem is to model the student's learning style, and deliver only those teaching media that are best suited to each learner's style.

1.3.2 Business Perspective

On the industry side of online education, the programs offering online education have grown in the past five years. Eduventures research firm [52] predicted that the online distance learning market would grow more than 38 percent in 2004, to reach \$5.1 billion in revenues. It also forecasts that student enrollment will top one million in 2005. These promising predictions also mean that the colleges offering online education will have to be competitive to take a share of the emerging market. Colleges and companies are competing in this area, with some of the companies partnering with colleges to offer online degrees (e.g.: Sylvan Learning Systems and University of Liverpool).

In the corporate area, companies need to continuously educate their employees about new technologies and changes in their business domain. Having an on-site instructor can be expensive in terms of labor costs and wasted employee work time. In addition, on-site instruction is time inflexible. Employees have to block out a time slot at

the expense of other important work. By using asynchronous online education, employees can work at their own pace and on-site or remotely. In addition, studying at one's leisure has the potential to break the routine of the daily continuous, focused work.

Production costs of online courses are high, but can be driven down by having a reusable course management system design that will minimize the marginal costs of each additional course. Since the Internet provides a simple means for deploying similar software systems quickly, reusable courseware design is intrinsic to online education.

1.4 Scope of the System

The primary objective of this thesis is to design a system to deliver course material that matches the student's learning style. To this end, the course material is preferably in a centralized course delivery system that includes meaningful descriptors about the content. A practical way of having these requirements is for the material to be present on a course website, with the content tagged via metadata.

Some online courses use the Web as the sole medium for course delivery, without the use of lectures, and some courses use the Web as a supplementary educational resource, thereby keeping the role of the teacher as the primary deliverer of course content. This thesis aims to deal with such supplementary education resources for face-to-face instruction, with the application test bed being the PIVoT system.

1.4.1 The PIVoT Project

PIVoT is the result of a collaborative venture at MIT between the Center for Educational Computing Initiatives and the MIT Physics Department. PIVoT houses a wide range of electronic materials that spans a classical mechanics class offered at MIT under subject number 8.01. This course, to date taught fairly traditionally, has an accompanying online site that offers many tutoring media and a large amount of information directly relevant to the course (<http://curricula2.mit.edu/>). The media pool that is presented to the student is varied and accommodates most learning styles accounted for in the educational literature. The course website offers recorded segments of lectures, the digital version of a textbook separated into topics and chapters, multiple-choice problems, simulations related to the studied subject, diagrams, charts, frequently asked questions (online static explanations, referrals to the textbook, or videos of sample problems solved by a professor), and discussion boards among peers, teaching assistants and the lecturer.

1.5 Goals

The goals of this thesis belong to two main domains: educational and computational. On the educational front, the course content delivered has to match the student's learning style at different points in time. On the computational front, the system has to be mathematically rigorous, architecturally sound, scalable and platform neutral.

This is achieved by the creation of a student learning style model and the delivery of learning material through a course management system, with the content and format delivery matching the student's changing needs at different points in time.

Logistically, the student is first taken through a dynamic questionnaire about activities in their day-to-day life. The system then makes inferences about his or her most probable preferred learning activities and creates an initial probabilistic student model. This questionnaire consists of a knowledge-based system (KBS) with questions, probability matrices for making inferences about students' learning styles, and a Bayesian updating system. After the questionnaire is used to create a model of the student, a multi agent system (MAS) can start delivering the course material that fits the learner's preferences. This MAS is formed of expert software agents that are each specialized in a specific domain of the learning experience. Since these agents have the knowledge about teaching along a specific LS dimension, they are called LS experts. Each of these agents recommends a ranked set of activities that it calculates to be most relevant to the student's needs. These recommendations are fed into a voting system, and the outcome is the aggregation of all the agents' recommendations. The student is then presented with the agreed upon list of ranked activities in terms of course content and format matching his or her learning style.

1.5.1 Educational Goals

Various researchers in the fields of cognitive science, psychology, and education have investigated different aspects of learning styles. While there is no agreement on a framework for assessing a learner's preferred style, the approach that this thesis adopts is

to consider the application field at hand and to apply the learning styles that fit the available media. Therefore, depending on the existing tutoring media, a different learning style model is applied. In this thesis, the adopted LS dimensions are those of perception, reception and processing of information, as well as the online interaction preference of the student with peers and faculty. The different learning style models are explained in detail in section 2.2, along with the different views of learning styles (psychological, behavioral or cognitive) that appear in the literature.

1.5.2 Computational Goals

On the computational side, the goal is to have an adaptive system that monitors the student's continually evolving learning style. The assessment of a student's learning style is a very delicate area because the functioning of the brain is not completely understood. In addition, our assessment of students' learning style is, in our work, limited by the types of information an online tutoring environment can reasonably collect. For instance, when an agent observes that the student is spending more than the average time on a course page, this might mean that the student is very interested in it, or it can be that the material is hard to understand, or the learner was simply away from the computer. The approach taken here is to assign a probability to the inference made from the student's answers to the LS assessment questionnaire and to the observations made about the student's navigation of the online course. The multi-agent system (MAS) also needs to have a rigorous negotiation methodology to aggregate the expert tutoring agents' recommendations.

1.5.2.1 MAS Architecture

This research aims to create a combination of a knowledge-based system (KBS) and an MAS. One sub-goal to this end was to create an Application Programming Interface (API) that simplifies the process of creating new tutoring agents. This stems from the need to make the system adaptable to the teaching media available and therefore to the learning styles dimensions considered.

The initial student model resulting from the KBS creates a profile of the student's learning style, and this profile is made available to the tutoring agents to start suggesting course material to the student. In an MAS, conflicts among the multiple agents' recommendations are likely and need to be resolved. An active area of research in MASs is negotiation, or how agents communicate, cooperate, and compete with each other [42]. This thesis presents an agent negotiation methodology based on *voting theory* and multi-criterion voting algorithms. Multiple algorithms are studied and assessed in the case of ranking multiple suggestions according to multiple criteria. This will be discussed in more detail in Chapter 4.

In MASs, a significant research area is the compromise between leading the student through the course material, and letting the student take control of the learning experience. Often today's pedagogical agents use interfering animated user interfaces that lead the learner through a learning path. For the case of supplementary online resource like PIVoT, Vogel et al. [121] mentions that it is important to allow the user to explore the Web resource according to self-directed goals. This research offers a midway solution, allowing the students and agents to communicate in a way that balances the students' ability to follow their own course of actions with the need for agents to guide

students along their proposed recommendations. The user is given the power to discard the agent's plan in favor of his or her own.

1.5.2.2 Scalability

The tutoring system design must support a knowledge base of hundreds of metadata informational tags about tutoring media, a considerable amount of knowledge about learning styles, probabilistic inferences about each student's learning style profile, and hundreds of students who can access the website simultaneously.

This thesis aims to achieve scalability in the knowledge base size, and handling of concurrent student logins. Over 500 students enroll in a typical fall term of MIT's introductory Physics course. At peak times, a significant number of these students may go simultaneously online, and the system must maintain acceptable performance.

1.5.2.3 Platform Neutrality

The online course content is open to outside users, and therefore the website is accessed from a wide range of clients. A requirement of this research is to deliver a consistent experience and interface across client platforms.

In addition, the server technology was chosen to allow portability to other platforms, as needed. Java was chosen as the platform of choice on the server side. Having such a server provides an approximately uniform experience to clients' platforms

and a reasonable assurance of functionality of tutoring agents, database management systems, mathematical computation, etc.

Chapter 2 – Literature Review

Due to the interdisciplinary nature of this research, a careful review of the literature spans several disciplines. Most of the current literature in Web-based pedagogy and intelligent tutoring systems comes in the form of case studies that combine various technologies and instructional design methodologies with various cognitive theories and pedagogical approaches. This chapter offers both a broad overview of the technological and educational theories and approaches used to create Web-based learning environments as well as a detailed review of the current research in learning styles.

This literature review begins with a discussion of the different adaptive hypermedia approaches in use. It continues with a discussion of learning styles theories and models. It concludes with a comprehensive survey of current intelligent multi-agent tutoring systems, which combine these technologies and theories in different ways.

2.1 Adaptive Hypermedia

Brusilovsky [24] introduced the different techniques of adaptive hypermedia. Adaptation can be done at two levels: link-level and content-level. The former is called *adaptive navigation support* and the latter *adaptive presentation*.

2.1.1 Adaptive navigation

2.1.1.1 Direct guidance

Direct guidance is the simplest technology of adaptive navigation support. It consists of the system deciding on the next "best" node(s) for the user to visit according to the user's goal and other preferences represented in the student model. Depending on the kind of system, it can be the goal of the work (in application systems), a search goal (in information retrieval systems), and a problem-solving or learning goal (in educational systems). In all of these cases the goal is an answer to the question, "Why is the user using the hypermedia system, and what does the user actually want to achieve?" Direct guidance was used in systems like Web Watcher [4] and HyperTutor [81].

2.1.1.2 Adaptive ordering

The idea behind adaptive ordering is to sort all the links of a particular page according to the user model and/or to some user-defined preferences: the more relevant link is placed closer to the top. This was particularly useful in information retrieval in the findings of Kaplan et al. [64] and Mathé and Chen [81].

2.1.1.3 Hiding

Hiding is the restriction of the navigation space by hiding links to irrelevant pages. A page can be considered as not relevant for two reasons: either mismatch in

student goal or disparity of capability. If the presented link is not relevant to the current goal of the user ([37], [114]) or if it presents material that exceeds the user's performance level or background ([24], [37]), hiding is a useful technique. In addition, hiding has the advantage of reducing the cognitive load on the users.

2.1.1.4 Adaptive Annotation

The idea of *adaptive annotation* technology is to augment the links with some form of comments that can tell the user more about the current state of the nodes behind the annotated links. These annotations can be provided in textual form [131] or in the form of visual cues using, for example, different icons [31], colors [22], font sizes [44], or font types [22]. The typical kind of annotation considered in traditional hypermedia is static (user independent) annotation. Zhao [131] discusses how link annotation is an effective way of navigation support in hypermedia.

2.1.2 Adaptive Presentation

Adaptive presentation is the adjustment of the content of a page to a particular user, according to current knowledge, goals, and other characteristics of the user. For example, a sophisticated user can be provided with more detailed and deep information, while a novice can receive additional explanations. Existing adaptive presentation techniques deals with text adaptation [14], [18], [24]. Text adaptation implies that different users may get different texts as a content of the same page. Adaptation to learning styles is an area that is still being researched ([14], [49]).

2.2 Learning Styles

Keefe [62] describes learning styles as the preferred ways through which learners interact with and process information in learning environments. Let us consider some of the earlier and more recent work in the field of LSs.

2.2.1 History of Learning Styles

Different views of learning styles were and are still discussed in the literature. Behaviorism, a reductionist view of human behavior, dominated the field in the first half of the 20th century. It was called reductionist because it used a black box approach based on empiricism, but such a simplified view left much to be desired. The idea behind behaviorism is that, in order to know what is going on in the mind, one observes solely overt behavior.

Jean Piaget [100] observed that children go through stages of development that have no relation to external stimuli, and the classical conditioning model alone could not explain these observations. In 1981, Mezirow [84] stressed the importance of processing and reflection of experiences in learning. Kolb considered learning as a cycle that goes through the stages of experience, reflection, and later action, which in turn becomes a concrete experience for reflection. He further refined the concept of reflection by dividing it into two separate learning activities, perceiving and processing [69]. Consequently, he added the “Abstract Conceptualization” stage. Learners ask questions

in the Critical Reflection stage and try to find answers in the Abstract Conceptualization stage.

Teaching in classrooms and online is still mostly essentialist, i.e. an approach that does not take learner experience into account. In addition, as Brookfield points out [21], teachers tend to be so concerned with presenting information that they overlook students' needs to reflect upon it. Rogers points out that "learning includes goals, purposes, intentions, choice and decision-making, and it is not at all clear where these elements fit into the learning cycle" [99].

2.2.2 Learning Styles Approaches

The literature on learning styles is vast, and different views of the learning process lead to different focal points of emphasis. The most studied and used learning styles inventories apply to learning and teaching in a classroom environment with face-to-face interaction and activities that happen outside the classroom space. Our situation is that of a web-based interface with the content presented on the course website and the student sitting at the computer and exploring it to learn the class material. With that said, the most used learning styles inventories include a variety of activities that are not applicable to an online environment. The different models that are widely accepted are presented below.

2.2.2.1 Dunn and Dunn Model

The Dunn and Dunn [35] learning style model traces its roots to two distinct learning theories: *Cognitive Style Theory* and *Brain Lateralization Theory*. Cognitive Style Theory is based on the idea that individuals process information differently on the

basis of either learned or inherent traits. Brain Lateralization Theory is based on the idea that each hemisphere of the brain assumes different functions: the left hemisphere achieves verbal and sequential abilities, while the right one is responsible for emotions and spatial holistic processing.

This model is based on five different categories of (1) environmental, (2) emotional, (3) sociological, (4) physiological, and (5) psychological conditions. For each of these categories, a number of elements contribute to the make-up of each individual's preferences. For example, the environmental category (or stimulus as called by Dunn) is affected by elements such as light, sound, temperature and furniture design.

They hypothesize that providing a variety of the learning methods, which accommodate diverse learning styles, can significantly increase learning success.

These stimuli are not meant to be applied to a website where the only inputs from the student are a keyboard and a mouse. This model is more suitable to a face-to-face interaction in a physical classroom environment.

2.2.2.2 Kolb Model

Kolb's *Learning Style Inventory* [70] is based on the works of educational theorist John Dewey [34], social psychologist Kurt Lewin [77], and developmental psychologist Jean Piaget [100].

Dewey [34] contrasted the traditional or teacher-centric teaching approach with that of the “new education”, or student-centric. His approach is supported by an underlying philosophy, which states “that there is an intimate and necessary relation

between the processes of actual experience and education,” therefore focusing on the need for learning to be grounded in experience.

Lewin’s experiential learning model consists of a concrete experience, from which observations and reflections are made, that lead to the formation of abstract concepts and generalizations, following which comes the testing of the implications of these concepts in new situations. The four phases are placed in a cycle, with testing leading back to experience. This shows his view of the continuous nature of experiential learning, which is the precursor to the Kolb Cycle.

Piaget's work was mainly devoted to the development of children’s intelligence. His findings suggest that the intelligence is not innate, but rather the result of interactions between the self and the environment. By testing the intelligence of children, he was interested in the reasoning process that the subjects go through. He noticed that at different ages, children were interested in different observations and activities. His four developmental stages are:

- Sensory-motor, from eighteen months to two years.
- Preoperational, from two to seven years.
- Concrete operations, from seven to eleven years.
- Formal operations, from eleven to fifteen years.

Piaget explains that learners’ ways of acquiring knowledge change “qualitatively in identifiable stages, moving from an enactive stage, where knowledge is represented in concrete actions and is not separable from the experiences that spawn it, to an iconic (preoperational) stage, where knowledge is represented in images that have an

increasingly autonomous status from the experiences they represent, to stages of concrete and formal operations” [100].

Drawing a parallel between Piaget’s theory and the Kolb cycle, the childhood learning stages will constitute the learning process of an adult in any learning process. The period spent before fifteen years of age learning from the world and internalizing observations represent for Kolb the constituents of a learning process that will lead the person from making observations to formulating abstract conclusions in a single learning cycle.

Kolb synthesizes the findings of Dewey, Lewin, Piaget and others to construct his own model of experiential education. His model is comprised of four phases that he locates in a circle, known in the literature as the *Kolb Cycle*. For complete learning to occur, one must proceed through all four parts of the cycle.

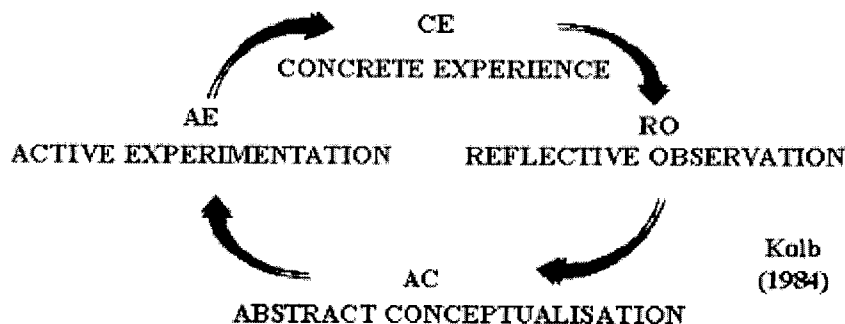


Figure 2-1: Kolb Learning Cycle

Learners don’t always go through the four stages, and they will have preferences on their individual ways of learning. The Kolb model (see Figure 2-2) classifies learners

as having a preference for (1) *concrete experience* or *abstract conceptualization* (how they receive information), and (2) *active experimentation* or *reflective observation* (how they internalize and process information).

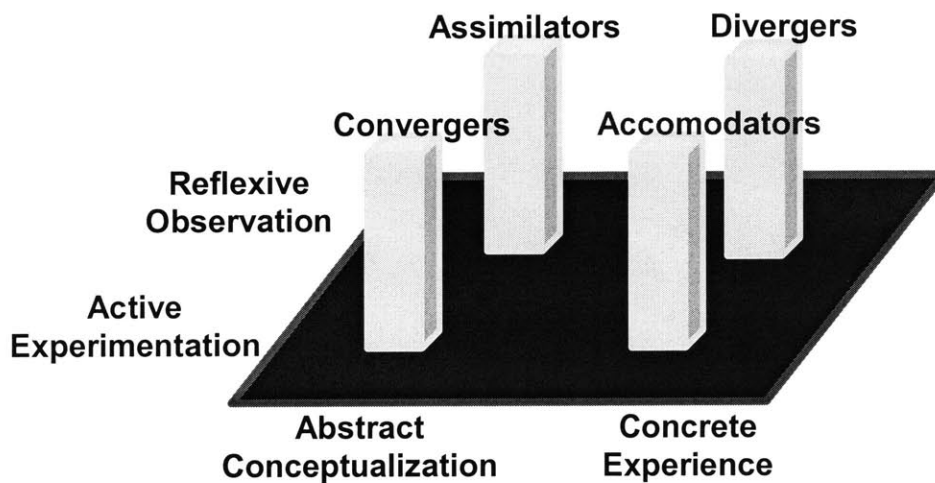


Figure 2-2: Kolb Model

2.2.2.3 Myers-Briggs Type Indicator

The Myers Briggs model ([87], [88]) was published in 1962. It was developed by Isabel Briggs-Myers, and it classifies students according to their preferences on scales derived from psychologist Carl Jung's theory of psychological types. Students may be:

- 1) Extroverts or introverts
- 2) Sensors or intuitors
- 3) Thinkers or feelers
- 4) Judgers or perceivers

This model has applications in a number of domains, but it is difficult and not very useful on a website to model the personality of the student. We would rather know about the learning preference and the cognitive style of the student.

2.2.2.4 4MAT Model

The 4MAT® model [82], which is based on the Kolb model, is constructed along two continua: perceiving and processing (see Figure 2-3). Human perception ranges between experience and conceptualization, while processing channels range between reflection and action. These elements are linked to form four stages of learning. Teaching the students concentrates afterwards on the interplay between left and right brain functions within the learning cycle: experiencing (left-brain), conceptualizing (right-brain), reflecting and applying (left-brain), and creating/acting (right-brain).

This teaching method is used to teach the students and is used as a guide in a variety of activities. This approach works well in a face-to-face setting with a large number of ways of interaction and activities. When applied online, the 4MAT model has some applications explored by Huitt [54]. According to him, the 4MAT system works well in an online environment that has been pre-designed according to the 4MAT model. However, in the case of learning resources developed without the 4MAT model as the designing criterion, the 4MAT teaching method would no longer be useful. Therefore, this model is not well suited to our domain of online, asynchronous learning.

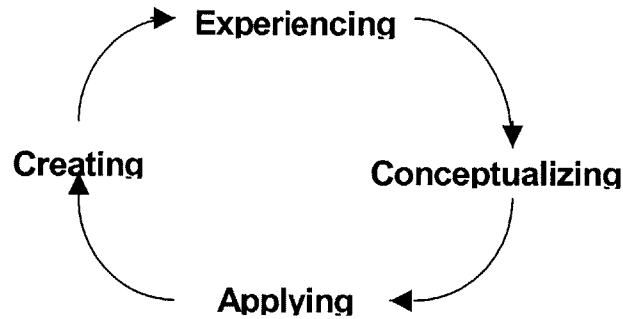


Figure 2-3: The 4MAT Learning Cycle

2.2.2.5 Reichmann and Grasha's Model

Reichmann and Grasha's model for motivation assessment is based on social interaction. This categorization indicates the likely attitudes, habits and strategies that students will take toward their work. It classifies students as independent v. dependent, avoidant v. participant and collaborative v. competitive.

These categories are well suited to a classroom setting, where the teacher can organize activities according to the preferences of the students. Our application field is to online learning and, therefore, assessing these categories would not be particularly helpful.

2.2.2.6 Gardner's Multiple Intelligences

Gardner defines intelligence as "the capacity to solve problems or to fashion products that are valued in one or more cultural settings" [41]. Gardner's multiple intelligences model classifies students along seven dimensions: (1) logical-mathematical, (2) linguistic, (3) spatial, (4) musical, (5) bodily-kinesthetic, and (6) inter- and (7) intra-

personal intelligences. Everyone possesses the seven intelligences at birth. Nevertheless, students will come into the classroom with different sets of developed intelligences. This set of intelligences that they have developed is their learning style in the context of classroom learning.

This model has wide application in a classroom setting. Teaching students who have different intelligences developed is a challenge to the teacher, and it may be time consuming for the students to go through the different ways of learning topics. In this sense, more individualized teaching is appropriate. The Internet may be one way of achieving this; but the lack of a human interaction, which is needed for categories such as interpersonal, intrapersonal and bodily intelligences is a major challenge. A subset of the seven intelligences is more suited to our context, and choosing this subset will be one of our tasks in assessing students' learning styles and teaching them accordingly. This will be covered in section 3.3.

2.2.2.7 Hermann Brain Dominance Instrument

The Hermann Brain Dominance Instrument is based on right/left brain research. The Hermann Brain Dominance Profile conceptually divides the brain into four quadrants (see Figure 2-4): left brained v. right brained and cerebral v. limbic. The upper left quadrant is labeled "A". He describes people with preferences in this quadrant as "logical, analytical, mathematical, technical, and scientific." The lower left, "B" quadrant describes people who are "controlled, conservative, organizational, and administrative." The lower right quadrant, "C" describes those who are "social, emotional, spiritual, and

talkative." The upper right, "D" quadrant is where creativity comes into the picture. People with preferences in this quadrant are "imaginative, synthesizers, artistic, and think non-linearly".

This categorization is a highly general one, and many factors enter into each of the categories. Breaking down these four categories into their characteristics helps tailor the content better to the student's needs, in order to teach them more directly.

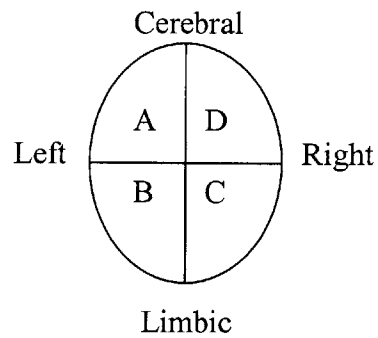


Figure 2-4: Hermann's Partitions of the Brain

2.2.2.8 Felder-Silverman Learning Style Model

Felder's [37] approach divides the learning continuum into well-defined parts that can be applied to a course website content delivery. Felder has separated the stages of learning into five well-defined levels. At each stage, students behave differently to attain their final goal of understanding.

This model classifies students as: (1) *sensing learners* (preferring concrete, practical activities, oriented toward facts and procedures) or *intuitive learners* (motivated by conceptual, innovative activities, oriented toward theories and meanings), (2) *visual learners* (favoring visual representations of presented material--pictures, diagrams, flow charts) or *verbal learners* (choosing written and spoken explanations), (3) *inductive learners* (learning better with presentations that proceed from the specific to the general) or *deductive learners* (following presentations that go from the general to the specific), (4) *active learners* (learning by trying things out, working with others) or *reflective learners* (studying better by thinking things through, working alone), and (5) *sequential learners* (following a linear, orderly progression, and learning in small incremental steps) or *global learners* (being holistic, systems thinkers, and learning in large leaps).

Felder also suggests corresponding teaching styles to each of the categories. These will guide the teacher in the content and the format delivery to the student. For example, if a student has an active learning style, he or she is presented with a problem-solving session; in the case of a reflective style, instructional material that emphasizes fundamental understanding is presented.

This model has tangible dimensions that can be measured in an online environment in order to assess the student's learning style. Paredes [95] chose this model for its direct link to adaptive environments. Similarly, Carver et al. [26] use Felder's inventory of learning styles for its practicality in online teaching.

2.2.3 Issues with Learning Style Models

More models than those discussed also exist and are used to a less significant extent than the ones that we described in section 2.2.2. Different models are not necessarily better or worse than others, but look at different views of learning, be they psychological, or behavioral, or physical, etc... Therefore, there is little basis to judge one approach being better than another. Rather, in our work, we assess each approach's applicability to our environment and our purposes. Moreover, there is no widespread agreement or rigorous evidence of any one approach over another leads to a better learning and teaching environment.

An additional issue with the different models is their creator's intent. The existing models were developed for classroom environments, rather than cyber environments where limitations exist, such as the difficulty of observing the students' attitude toward the material, and where other communication ways are adopted (mouse, keyboard, etc...).

The models that have been developed have common categories, such as the sensing/intuitive category present in both Felder's model and the Briggs-Myers Type Indicator.

Since the models that were developed were intended to be applied in the classroom environment, some categories within the models are not applicable to distance education. For instance, in the Myers-Briggs Indicator, one of the learning style dimensions in that model is extrovert/introvert. This can be observed in a classroom environment by noting the level of participation of the students and their involvement in group activities. However, this is not applicable to an online environment, where live

participation is not typically present. In this case, we will have to disregard the extrovert/introvert category from the model and focus on more tangible aspects relevant to the field of application.

With that said, the approach of adopting a specific model would be hindered by such limitations. Therefore, this thesis takes a new approach to adapt learning styles to the online environment. This approach will be explained in Chapter 3.

2.3 Artificial Intelligence in Education

Part of the challenge of artificial intelligence includes the challenge of defining it. It is difficult to define artificial intelligence (AI) without semi-circular reasoning about what is “intelligence” itself. Nilsson [90] acknowledges this in defining AI broadly as the discipline “concerned with intelligent behavior in artifacts.” Winston [123] more directly defines artificial intelligence as “the study of the computations that make it possible to perceive, reason, and act.” He continues by distinguishing AI from the related disciplines of psychology and computer science, due to AI’s emphasis on computation over the former, and perception and reasoning over the latter.

Although AI has already produced practical and useful systems, the ultimate goal of achieving human-level intelligence is still quite far away [90]. That being so, there is strong debate among researchers about the best approach to AI. Over the last half-century, many different paradigms have emerged.

The views and definitions of AI can be summarized into the different views about AI perception and performance measurement. Table 2-1 presents a simple matrix that

explains those two dimensions. A system has the ability to be intelligent by a combination of an intelligent expression which is *perceived* and that can be *measured* against pre-defined performance levels.

“Perception” in artificially intelligent systems spans a continuum between the *behavior* of the system and the *reasoning* process adopted by the system. A system’s intelligence is “measured” against expected *ideal performance* or *human-like performance*.

Each of the four AI schools considers a combination of perception and performance measurement to assess the intelligence of a system (see Table 2-1).

Table 2-1: AI Systems

		Thought/Reasoning			
Human-Like Performance	Turing Test <i>Systems that think like humans</i>	Laws of Thought/ Logic <i>Systems that think rationally</i>		Ideal Performance	
	Cognitive Science <i>Systems that act like humans</i>	Rational Agents <i>Systems that act rationally</i>			
		Behavior			

There are various models of AI that have been applied to education systems. This section offers an overview of models used in AI that have strong applicability to education and intelligent tutoring systems. The models discussed are the ones used in the LS system that is the subject of this thesis. Expert systems use classical AI to draw

inferences from knowledge in a specific domain. Probabilistic Inference and Bayesian Networks are used to model uncertainty. Intelligent agents, specifically pedagogical and collaborative agents, work together to provide a human-like group of tutors.

2.3.1 Expert Systems

Expert systems, or knowledge-based systems, apply classical AI reasoning techniques to facts and rules about a specialized field. Expert systems contain four key components: a *knowledge base* consisting of logical rules about a domain, a *knowledge-acquisition* system which propagates that knowledge base with these rules, an *inference engine* which uses logic and the predicate calculus to apply these rules to given conditions, and an *explanation system* to communicate the ideas and logic to a user.

Knowledge engineers work with experts in the domain by asking them about situations in search of similarities between them. As such, two key heuristics enable knowledge engineers to acquire knowledge: first, be as specific as possible, and second, look for outwardly similar appearing situations and find distinguishing characteristics among them [122].

A survey of applications of expert systems highlights domains such as medicine, engineering, and business [111]. The canonical example of an expert system is an automatic medical diagnosis system (e.g. MYCIN). Expert systems are used in education because they can be used to assess students' knowledge and help detect misconceptions in students' logic [101]. They can also be used to infer the personality, intelligence and preferences of students. This area has not been well researched, and therefore a part of this thesis presents an expert system that assesses the learning style of students.

2.3.2 Probabilistic Inference and Bayesian Networks

Often artificially intelligent systems only have uncertain information about the task at hand and their environment. As such, probability theory is frequently used in the design of AI systems. Often, the certainty of one unknown affects the certainty of other unknowns. Additionally, new information about the task or environment, which may itself be uncertain, alters existing (a priori) probabilities to create a new (a posteriori) view of the world. This kind of reasoning with uncertainty is known as *probabilistic inference*, or *Bayesian inference* in reference to Bayes' theorem, which relates a priori to a posteriori probabilities.

Bayesian networks are convenient structures for representing probabilistic inference. Bayesian networks are also referred to as *belief networks*, because they can be used to represent an intelligent system's uncertain "beliefs" about the environment in a formally correct way. A Bayesian network is a directed, acyclic graph (DAG) where each node represents an unknown. Edges in the graph represent causal relationships between unknowns, and thus such networks are sometimes called *causal networks*.

Bayesian networks offer a mathematically formal alternative to less formal "fuzzy logic" models found in artificial intelligence. Bayesian networks have pedagogical applications because of their use in modeling student knowledge and profile. Due to the inherent uncertainty of assessment and other data gathered by a tutoring system, belief networks are quite popular in intelligent tutoring system design.

2.3.3 Software Agents

Intelligent systems and agent-based systems are new paradigms for developing software applications [55]. While agent-based systems are of intense interest to researchers in computer science and artificial intelligence, the terminology used to describe these systems, including the definition of the word *agent* itself, is not universally agreed-upon [20]. Nevertheless, a workable definition for agents and agent-based systems is presented here, along with a vocabulary for understanding agent-related concepts and characteristics.

A software agent can be thought of as any entity capable of performing tasks on the behalf of a user or contracting party in which intelligence and specialized knowledge of a domain is required [123]. Although this definition could be used to describe human and hardware agents (i.e. travel agents, robots), emphasis here is given to intelligent software agents. As such, unless otherwise stated, the terms *agent*, *intelligent agent* and *software agent* will be used interchangeably. An alternative definition for agent is any system capable of interacting independently and effectively with its environment via its own *sensors* and *effectors* [75].

An agent-based system is any system in which the key abstraction in its design is an agent as defined above [56]. It is important to note that agent-based systems may contain any non-zero number of agents. The multi-agent case is intuitively more complex, due to the issues regarding *negotiation* between agents, but preferable for certain problems. These systems are referred to in the literature as multi-agent systems (MAS). The single-agent case is also common and often adequate for many tasks, such as the class of systems known as *expert assistants*.

2.3.3.1 Characteristics and Classification

Agents can be best understood through the fundamental characteristics that describe them. First, agents are situated in an environment (e.g., the Internet) with which they can interact *responsively*, *proactively* and *socially*. Agents are *responsive* in that they can perceive their environment, detect changes within it, and take appropriate actions in a timely fashion. Agents are *proactive* in that they do not simply react to changes in the environment, but rather take the initiative when appropriate, demonstrating goal-directed *behavior*. Agents are *social* in that they interact with their users, other agents, or both to solve problems and to achieve their goals.

Agents and agent-based systems are often classified and categorized in several ways, yet these distinctions are not universally agreed on [91]. Agents are best categorized and analyzed multi-dimensionally; this section offers both the fundamental properties of agents, and common characteristics used to provide a broad understanding of agent research.

2.3.3.1.1 Deliberative and Reactive Agents

Agents are described as being either *deliberative* or *reactive*, depending on the presence or absence of an internal symbolic reasoning model, respectively. Deliberative agents assume an explicit symbolic representation of their environment, maintaining the traditions of classical AI. Reactive agents have their roots in the more decentralized

nouvelle AI and provide increased dynamic flexibility at the expense of complex reasoning processes.

Deliberative agents normally create a model of their environment in advance, and their model becomes the main component of the agent's knowledge base. Creating an elaborate internal model that achieves an adequate amount of functionality requires a careful design and ample resources. Additionally, one must be able to anticipate at design time errors and irregularities that might occur. Not surprisingly, deliberative agents have only limited suitability for use in dynamic environments [130].

In addition to their internal symbolic model of their environment, deliberative agents have the ability to make logical decisions from knowledge stored in their internal state. Such agents have symbolic representations for beliefs, desires, goals, intentions and plans, and architecture designed to use this "mental model" to make appropriate decisions about which actions (if any) to take. Often these agents follow so-called BDI architectures, short for *beliefs*, *desires*, and *intentions* [79].

Deliberative agents have a range of problems, often reflecting the problems of classical AI itself. The chief criticism of deliberative agents comes from their rigid structure. Software agents, especially those situated on the Internet, have too dynamic of an environment to make effective use of a symbolic model designed before runtime. Often a workable model of a complex situation (such as the relationship between an agent and its "real" environment) cannot be fully understood at design time. In addition, adding this data at run time is difficult as well. As stated by Zarnekow and Wittig [130], "because the necessary knowledge and the required resources are not normally available,

it is difficult for such agents during their execution to add in their existing model new information or knowledge about their environment.”

Reactive agents stand in sharp contrast to, and came as a reaction to the philosophy of, deliberative agents. Lacking an internal symbolic model of their environment, these agents obtain their intelligence and behavior not from a centralized, complex reasoning process based on internal knowledge, but from the complex interactions of simple stimulus-response rules [91].

Reactive agents need not have a complex structure to navigate a complex environment [125]. Instead, reactive agent designers observe simple principles or dependencies, producing task-specific *competence modules* that are easy to design and correctly implement. This decentralized approach serves to increase the fault tolerance and robustness of reactive agents, making them highly suitable for dynamic environments such as the Internet.

Reactive agents are not without their own set of problems. It is much harder to make reactive agents demonstrate goal-oriented behavior. Purely reactive agents also do not possess any capabilities to create plans. Additionally “the extent to which [the inability to create plans] negatively affects its tasks to be performed cannot be fully determined [130].”

It is important to realize that modern AI systems, including the ones mentioned later in this review, are neither purely deliberative nor purely reactive. It is common to design hybrid systems that leverage the advantages of both approaches and integrate them into a common platform. The LS teaching system is an example of such a hybrid system.

2.3.3.1.2 Static and Mobile Agents

Agents may be classified as either static (stationary) or mobile. Mobile agents have the ability to move freely around the network in which they reside, whereas static agents are bound to a single computer. Static agents may send messages to other agents via a network, but each agent runs exclusively on one machine. Mobile agents can clearly pose a security risk in open networks such as the Internet and are uncommon in the literature on pedagogical agents. Thus, stationary agents are emphasized here.

2.3.3.2 Collaborative Agents

According to Nwana and Ndumu [91], *collaborative* agents emphasize autonomy, as well as cooperation and negotiation with other agents, in order to perform tasks for their owners. Collaborative agents aim to solve problems too large for a centralized agent system or to allow for the interconnection of existing legacy systems. These agents tend toward large, static multi-tasking agents. While these agents have applications in organizational decision-making or industrial settings, they are not seen as applicable to pedagogy and as such are not emphasized in this review.

2.3.3.3 Interface Agents

Interface agents emphasize autonomy and learning in order to perform tasks for their owners [91]. These are agents used in simplifying, enhancing, and assisting user interactions with existing applications; and thus interface agents adhere to the metaphor of *personal assistant* [80]. Interactive agents provide proactive assistance by observing and monitoring a user's actions and suggest better ways of doing the task. Some

examples include the popular Microsoft Clip Assistant, and course assistants such as ADELE [60].

Maes [80] observes the two key preconditions necessary for creating adequate interface agents: first, that there exist repetitive behavior (necessary for the agent to learn from) and second, that this behavior varies for different users. Many pedagogical applications where agents are useful meet these preconditions.

Interface agents vary in the degree of visibility and intrusiveness. This research has opted for non-animated agents that interact textually with the user, provide the user with a recommendation upon request, and have the option to be turned off.

2.3.3.4 Pedagogical Agents

Pedagogical agents are software agents used for teaching or tutoring purposes. Animated pedagogical agents are a common sub-class of *embodied* pedagogical agents. Embodied *pedagogical agents* have anthropomorphic traits, including emotion, natural language ability and personality. Animated pedagogical agents display their embodiment through a cartoon-like interface [80], [114]. Since they are often used in tutoring the user, this application of agent technology is discussed at length in section 5.1 on Intelligent Tutoring Systems (ITS).

2.4 Intelligent Tutoring Systems

The field of Intelligent Tutoring Systems (ITS) is intrinsically multi-disciplinary, requiring research efforts in several disparate domains [62]. Computer science provides insight into the computational aspect of ITS. Cognitive science reveals information on

cognition and the epistemological issues of knowledge. Behavioral psychology helps to understand learners' behavior. Educational studies investigate the effectiveness of different pedagogical approaches. This section offers a brief history of early tutoring systems, followed by a categorized review of the current case studies in ITS design and practice.

ITSs have several common components, usually found in all such systems [25]:

- *Student model*: The progress and the evolving interests of the student are continuously measured throughout a session or across multiple sessions.
- *Knowledge base*: Information and rules about the domain are stored as resources for the ITSs.
- *Evaluation module*: Computation of the next ITS action is based on the understanding of the student's performance and interests.
- *Pedagogical rules*: Pedagogical approaches vary greatly across intelligent tutoring systems. The rules translate the adopted pedagogical strategy into the ITS representation.
- *Learning theory*: Different theories approach learning from various directions, including behaviorism, cognitivism, and constructivism.

Most of the conceptual foundations used in today's tutoring systems were developed before 1990 [25], with current systems expanding and combining seminal principles from these early intelligent tutoring systems. The following sections analyze the aspects of intelligent tutoring systems research that are of relevance to this thesis.

Each section offers both the seminal work in that area as well as a survey of current research on that aspect.

2.4.1 Learner Modeling

In the early 1980's, a tutor for the programming language LISP was created at Carnegie Mellon University, based on the ACT* model of cognition [28]. ACT* identified *procedural* and *declarative knowledge* as part of its expert problem-solving model. ACT* was seminal for introducing the *model tracing* paradigm, whereby the system responds with hints to divergences students make from the path of the expert model. Today, several systems use model tracing to control student-tutor interactions ([28], [96]).

Model tracing, while an effective approach to creating a model of the student's knowledge, is an extremely difficult task due to the search spaces involved [11], since the number of paths one can navigate an educational system grows with time. Model tracing is better suited for narrow-domain courses. Additionally, assessment data is often scarce and inadequate to make certain determinations about student knowledge in particular concepts. There are several approaches to solving this problem, including Bayesian networks, other probabilistic approaches, fuzzy logic, etc.

2.4.1.1 Probabilistic Models

In 1987, Tennyson and Park produced the Minnesota Adaptive Instruction System (MAIS) [28], which uses the systems approach to instructional design. MAIS adapts the

topics students encounter using Bayesian predictive statistical models. The difficulties of the questions posed are adjusted based on students' performance. Extensions based on Bayesian approaches are common in the literature today ([85],[86]).

Millán [85] proposed a more advanced adaptive approach to minimize intervention by knowledge engineers. Murray [86] created an easily implemented linear-time algorithm for Bayesian student modeling that can easily be used by ITS developers without deep understanding of Bayesian statistical analysis.

2.4.1.2 Non-Probabilistic Models

Alternatives to Bayesian network student models mirror alternatives in AI in general. Fuzzy logic has shown promise in some designs [117], whereas information theory is used in others [115].

Another approach to student modeling shifts focus from approximating the level of understanding of each student to understanding the way in which students utilize the learning system, improve and refine it, and thereby enhance instructional material delivery.

2.4.2 Cognitive and Pedagogical Approaches

There are many different cognitive and pedagogical approaches to ITS design. They address issues of student versus tutor control, how to apply cognitive theories of learning, and the role of assessment in ITS. As stated above, there is great divergence in cognitive and pedagogical approaches. This section offers a short review of recent work.

Several studies conducted in recent years have revealed that introductory physics students have more difficulty solving qualitative problems than quantitative ones. In order to improve the instruction of the subject, Albacete and VanLehn [2] created the Conceptual Helper, part of the Andes project at the University of Pittsburgh. The Conceptual Helper follows the model-tracing paradigm. That is, the tutor attempts to model the thought process of the student as he or she steps through the problem. While the system has been shown to be successful, it is strongly tied to the unique problems of physics education. One of the main goals of the Conceptual Helper is to solve common misconceptions about physics; while there is a lot of literature to support this approach, it is not well suited to use beyond physics.

The level of intelligence in ITSs is also a matter of great debate. Alevan and Koedinger [67] studied the use of intelligent and “dumb” help systems in a ninth grade geometry ITS and found that students often lack the meta-cognitive awareness to seek help when needed. While they suggest forcing students to use help even when they do not solicit it, other researchers emphasize the importance of student control in ITS use, to avoid information overload or disorientation [118].

On the level of intelligence in the tutors, Baylor [8] observes, “More intelligence is not necessarily better from a pedagogical perspective.” Boulay [18] shows that, while unintelligent versions of educational software produce marked improvements over traditional pedagogies, the intelligent versions of the same software improve results more dramatically. Furthermore, increasing the knowledge of the tutor decreased the number of steps necessary to solve problems. Boulay notes, however that far more time is spent on

intelligent v. non-intelligent versions, which may not only bias the result but also be a strong disincentive in time-pressed educational settings.

2.5 Conclusion

The areas of education, learning styles, artificial intelligence, expert systems and multi-agent systems have been well researched and developed. However, the combination of these areas in web-based education has not been approached previously.

Our goal is to combine these areas, evaluate their application to an online educational environment, and design a system that incorporates user profiles into web-based delivery systems. Our contributions will be at the technical, technological and pedagogical levels.

Chapter 3 – A Knowledge Based System for Learning Style Assessment

When information about a topic is presented in different formats and involves various activities, different students are interested in diverse ways of learning due to their individual learning styles. For instance, visually oriented students may benefit more than others from viewing videos and interacting with simulations. In order to make use of this idea in online learning environments, we need a procedure to assess the likely learning styles of students.

In this chapter, we present the approach that was adopted to assess an initial model of the student's learning style. This approach consists of a knowledge-based system applied to the assessment of learning styles. Each student's membership in a learning style category (e.g. visual/verbal/neutral) is estimated probabilistically. We estimate this probability value by using a sequential Bayesian approach to administer a dynamic length questionnaire that aims to attain a desired confidence level estimate while asking a minimal number of questions. This minimal number is the result of the minimization of the penalty attached to each question asked.

The system that was designed and developed is part of a larger undertaking, which is to teach the online students according to their learning style profiles. This questionnaire will be used as a starting point for a multi-agent system (MAS) that will rely on this initial LS assessment to start delivering the teaching media that best fit the student's profile. Our definition of an agent is an entity that is independent, proactive, knowledge-ridden and information acquiring, task-oriented, and aware of changes in its environment [57], [124]. The students' initial learning style profile will be used as the basis for a model that will be continuously updated while the student browses the course website. Student actions will trigger tutoring agents' action of updating the student profile. This in turn will affect agents' decisions on the best matching teaching material. This is a way of having the MAS adapt to students' preferences and teach them according to their best current estimates of each student's learning style profile [102].

3.1 Introduction

Popular and commercially available learning style (LS) assessment tests generally consist of a fixed number of questions and compute the sums and averages of all the questionnaire answers. This approach has drawbacks both in terms of the time spent by the student answering questions and in the accuracy of the results. If the static number of questions is large, it might lead the student to answer the questions hastily and therefore erroneously. In addition, the LS tests available do not provide a measure of the confidence of the results returned. The error margin of the assessment results is not provided, and this may mislead the user about the result's accuracy. Some available assessment tools are the Solomon and Felder Index of Learning Styles [37], the Fleming VARK questionnaire [39], and the Kolb Learning Style Inventory (LSI) [70].

We have designed and developed a knowledge based system (KBS) that consists of a dynamic questionnaire, meaning that the number, order and content of questions vary from one run to another, depending on the user's answers. This system inquires about the student's out-of-classroom experiences and preferences, makes inferences about the student's belonging to a particular LS profile, and provides the confidence level of the assessment.

The application field that we chose is a physics classical mechanics class offered at MIT under subject number 8.01. This course, while taught fairly traditionally, has an accompanying online site that offers many tutoring media and a large amount of information directly relevant to the course (<http://curricula2.mit.edu/>). A more detailed explanation is given in section 1.4.1. The media pool that we present the student with is varied and accommodates most online learning styles accounted for in the literature. The course website offers recorded segments of lectures, an electronic version of a textbook separated into topics and chapters, multiple-choice problems, simulations related to the studied subject, diagrams, charts, frequently asked questions (online static explanations, referrals to the textbook, or videos of sample problems solved by a professor), and discussion boards among peers, teaching assistants and the lecturer.

In this chapter, we introduce the available LS assessment tools in section 3.2. This will be followed by our LS approach, its added value, how it maps learning to teaching style, and its advantages in accounting for different LS models (Sections 3.3 to 3.5). The KBS content and design is described in section 3.6, and the methodology we used to get to the assessment results in section 3.7. The results of running the system are discussed in section 3.8.

3.2 Learning Styles Assessment Tools

An overview of some available assessment tools demonstrates the need and benefits of a KBS for assessing learning styles. The Soloman and Felder Index of Learning Styles [37], the Fleming VARK questionnaire [39], and the Kolb Learning Style Inventory [70] have advantages and drawbacks in terms of effectiveness, reliability and efficacy.

3.2.1 Soloman and Felder Index of Learning Styles Questionnaire

The Soloman-Felder Index of Learning Styles (ILS) [37] consists of a self-contained module for assessing student learning styles and has wide applicability. The learning style dimensions that the ILS considers are detailed in section 2.2.2.8. After taking the survey on-line, students receive instant results in the form of a profile of their dominant learning styles. The results page has links to Felder's website, which provides additional information on learning styles. The questionnaire is composed of a fixed list of forty-four binary-choice ("a" or "b") questions.

The results returned compute the average of the answers. The accuracy and standard deviation of the results is not made available. Zywno [133] collected ILS responses for several hundred engineering students and assessed test-retest reliability, internal consistency reliability, and several quantities related to the independence and

construct validity of the four instrument scales. She concluded that the ILS meets criteria of acceptability for instruments of its type.

Figure 3-1 shows an example of the results of the questionnaire. The “X” shows the positioning of the subject along the active-reflective LS dimension. The significance of the numbers is not clearly explained. However, this shows that the student has a slightly more active than reflective style. ACT and REF stand for *active* and *reflective*.

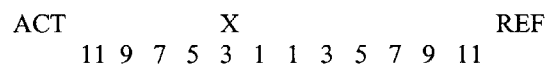


Figure 3-1: Soloman-Felder ILS Example Result

3.2.2 The VARK Learning Style Inventory

The VARK inventory introduces students and teachers to a variety of different approaches to learning. The acronym VARK stands for Visual, Aural, Read/write, and Kinesthetic sensory modalities that are used for learning.

The VARK questionnaire consists of twenty-three four-part questions. When scored, it provides the averages of the answers and classifies the user into one or more of these learning modalities.

3.2.3 Kolb Learning Style Inventory

The Kolb Inventory [70] is a 12-item assessment tool developed by David Kolb. Based on Experiential Learning Theory, it identifies four preferred learning styles:

- Diverging: Combines preferences for experiencing and reflecting
- Assimilating: Combines preferences for reflecting and thinking

- Converging: Combines preferences for thinking and doing
- Accommodating: Combines preferences for doing and experiencing

The first version of the LSI was released in 1976. A study by Ferrell (1983) showed that it was the most psychometrically sound among four learning instruments of that time. In 1985, version 2 of the LSI was released, but the studies showed a poor test-retest reliability measure ([112], [118]). However, a study by Veres *et al.* [118] showed that randomizing the order of the LSI version 2 items results in a dramatic improvement of reliability. The latest LSI version 3 revision has significantly improved psychometric properties, especially test-retest reliability [70].

However, the inventory asks the subjects questions directly targeting their learning styles, such as:

When I learn:

___ I like to deal with my feelings

___ I like to watch and listen

___ I like to think about ideas

___ I like to be doing things

By asking these questions, the tester assumes that the student is aware of his or her learning style, and the inventory becomes a paraphrase to the students about what they already know. Consequently, this approach differs from our current approach, in which the questions focus on out-of-classroom experiences in order to infer learning preferences.

3.3 Learning Styles Approach

Wolf [124], who worked on learning style-based e-learning, mentions that “it still remains unclear what aspects of a learning style profile are worth modeling and which is the most effective approach for a particular style.”

We propose a solution to the limitation listed above. The approach is to choose only the LS dimensions applicable to an online learning environment, have the agents observe the students’ preferences along them, and suggest to the students the teaching media that match their style. By achieving this, the online teaching site will account for different aspects of the LS models already developed and be independent of any particular learning style model.

The defining criterion that we adopted for choosing the applicable LS dimensions is to select the ones that can be used to describe the material or activity available at hand. We have used the LS dimensions below as a *descriptor* and a *comparator* of the educational material. For example, a recorded lecture video can be *described* at the perception (sensing/intuitive), input (visual/verbal), processing (active/reflective) and interaction (individual/group) levels. The LS dimensions were also used to *compare* the educational media: a problem set is a medium that involves *more* ‘activity’ than a video segment does.

The dimensions that form our model also provide us with a way of reconstructing different models or parts of the LS models that we introduced previously. For example,

we can reconstruct the Kolb model from the perception and the processing levels (see Table 3-1). Other models, such as the Myers Briggs Type Indicator, have dimensions that are applicable to a website setting, such as the perception level.

3.4 Learning Style Dimensions

Table 3-1 presents the different LS dimensions, the corresponding categories, and their defining criteria. Most of these categories overlap with other models that were discussed in section 2.2.2. In addition to the learning styles seen so far, we will consider the individual/group oriented LS. This addition stems from the availability of the ‘discussion board’ teaching medium in online environments. The chosen dimensions proved to be easily observable and applicable to the online course material. If we were applying the system to an environment where different teaching media were available, then other dimensions would be added, deleted or modified. For example, if we had a synchronous virtual classroom interaction between the teacher and the students, then the categories avoidant and participant from the Reichmann and Grasha’s model [103] might be added to the LS profile.

The criteria defining the different learning styles will be the subjects of the questions that the KBS will inquire about. For example, in assessing the sensing/intuitive style, the questions will be related to the time spent on tasks, the level of activity involved, and the nature of the content.

The different LS dimensions that were chosen to be most useful in the available online environment are explained below, and are summarized in Table 3-1.

3.4.1 Perception Styles

The learning experience starts with the learner's *perception* of the material. At this stage, the learner is either more *sensory* (sensing) or *intuitive*. *Sensors* concentrate on information gathered through the five senses. They are interested in "just the facts" that they need and do not want to be bothered with any information or ideas that may confuse the issue. Alternatively, *intuitive* students are much more interested in meanings and relationships than they are in the facts themselves. They are very good at reading between the lines and tend to anticipate future events. This dimension can be measured by the time spent, the level of activity involved, and the content's nature (theory or application).

3.4.2 Reception Styles

Learners *receive* the information through two primary channels: visual and auditory. *Visual* learners remember best what they see (pictures, diagrams, flow charts, time lines, films, and demonstrations). *Verbal* learners benefit more from words (written and spoken explanations). However, everyone learns more when information is presented both visually and verbally. This dimension can be measured by the format of the teaching material and the activity it involves from the student.

3.4.3 Processing Styles

At the *processing* stage, *active* learners tend to retain and understand information best by doing something active with it (discussing, applying it or explaining it to others). They have a tendency to test and spend time experimenting with simulations, changing values of variables and observing the results. In addition, *active* learners tend to like group work.

On the other hand, *reflective* observers spend more time on the theoretical aspects of a subject to try to understand it thoroughly. They are motivated by reading the textbook, analyzing a diagram or a chart, and spending more time on the material than their active counterparts.

3.4.4 Interaction Styles

The activities that the student is exposed to vary between individual and group work. In online environments, most of the work is targeted towards individual activities. However, in the case of PIVoT (see section 1.4.1), students can interact with peers and teaching staff, and therefore have the ability to exchange ideas in a group setting. We therefore add a fourth learning style dimension, called *interaction* (see Table 3-1).

Table 3-1: Learning Styles and Corresponding Observed Criteria

Learning Style Dimension	Learning Style Categories	Observed criteria
<i>Perception</i>	Sensing/ Intuitive (SI)	- Time spent - Content's nature - Kinesthetic activity
<i>Reception</i>	Visual / Verbal (VV)	- Format (text, video...)
<i>Processing</i>	Active/ Reflective (AR)	- Kinesthetic activity - Material reviewing
<i>Interaction</i>	Individual/ Group (IG)	- Undertaken activity - Interaction level

3.5 Teaching Style Dimensions

Each of the above-mentioned learning styles corresponds to a teaching style. Arguments abound as to the need to teach the students in their preferred style or to show them ways of exploring the material in modes different from their preferred style. Borg and Shapiro [17] show that a match between a student's and the professor's MBTI classifications results in improved student performance. Ziegert [132] found that having a personality type that matches the professor's had no significant effect on a student's grade in upper-level economics.

In our case, we are dealing with a website that is not the sole learning resource for the course, but rather is a supplement to the class lecture and discussions. Therefore, we are strong proponents of teaching the students according to their preferred LS and leaving other ways of exploring teaching to the physical classroom where the teacher has to account for multiple learning styles. By exploring different facets of presenting the course material in the classroom, the teacher will achieve both the goals of teaching according to the style of the students and of exposing them to other ways of receiving and processing the information presented.

The teaching styles corresponding to the learning styles of Table 3-1 are summarized in Table 3-2.

Table 3-2: Corresponding Teaching-Learning Styles

Teaching Style Dimension	Learning Style Categories	Teaching Style Categories
<i>Perception</i>	Sensing	Concrete
	Intuitive	Abstract
<i>Reception</i>	Visual	Visual
	Verbal	Verbal
<i>Processing</i>	Active	Active hands-on
	Reflective	Passive
<i>Interaction</i>	Individual	Individual learning
	Group	Group activities

3.5.1 Perception Level

Teaching sensing people is effective by delivering *concrete* information (facts, data, real or hypothetical experiments and their results), problems to solve, explicit illustrations of observation of surroundings, empirical experimentation, and concrete examples of the phenomena the theory describes.

As for intuitive subjects, the tutoring system should present *abstract* concepts (principles, theories, mathematical models), material that emphasizes fundamental understanding, open-ended problems and exercises that call for analysis and synthesis.

3.5.2 Presentation Level

For *visually* oriented students, the teaching should focus on presenting pictures, schematics, graphs, charts, and show video segments of the lecture.

Verbal students will be referred to textbook parts, lecture slides, written explanation, frequently asked questions, and problem sets.

3.5.3 Processing Level

Active style learners look for material that emphasizes practical problem-solving methods and hands-on demonstrations. Tutoring systems should also provide opportunities for students to do something active by suggesting small-group activities. In addition, drill exercises and problems for the practice of the basic methods being taught greatly benefits this category of learners.

Reflective students prefer materials that emphasize fundamental understanding. The tutoring system should wait for the student to ask for more information, instead of pushing it to him or her; this will give students the opportunity to think about the material at hand. Drill exercises should not be given priority, but reflective learners welcome open-ended exercises that call for analysis and synthesis.

3.5.4 Interaction Level

Students with an *individual* style can be left to perform common online teaching activities, be they problem sets, videos of lectures, textbook referrals, etc. *Group-oriented* individuals should be encouraged to use discussion boards, where they can discuss theoretical and applied course material with their peers. In addition, questions are encouraged, and students can pose queries to the teaching assistants in a manner so that their peers benefit from the answers as well. The system should also direct them to the section of posted frequently asked questions related to the studied material at hand.

3.6 Expert Learning Style Assessor

This section will discuss the requirements needed for the KBS Learning Style Assessor. The nature of the questions, the attributes of the expert system, and the need for a Bayesian updating system are discussed in detail. The subject of student profiling and classification into a learning style category is also introduced.

3.6.1 Overview

The goal of the system is to determine a student's LS through an adaptive questionnaire. The purpose of this section is to analyze the needs of expert systems, and particularly the specifics of this KBS. Previous KBSs were typically built to capture expert knowledge and suggest informed decisions, or to perform a diagnosis in a scientific setting (e.g., MYCIN [111]). In contrast, our system performs a qualitative assessment about human personalities and preferences or more precisely, their learning styles. Therefore, it will rely on psychology and judgment where matters start diverging from rigorous hard science and mathematically proven theories. Due to that reason, probabilities play a major role in quantifying doubt about statements and their inferences.

3.6.2 Importance of the KBS

The system that we are proposing is a *dynamic* questionnaire, meaning that the number, order and content of questions vary from one run to the other depending on the candidate's answers. This KBS uses a *sequential Bayesian inference* method, which relates a priori to a posteriori probabilities. After getting an answer A_{ir} from the candidate ('yes', 'neutral' or 'no'), we use Bayes' rule to compute the posterior probability $p[C_{jk}|A_{ir}]$ of the LS dimension k . A_{ir} is the answer to question i , with r being the response.

C_{jk} represents type j for LS dimension k . The learning style dimensions in our case are SI, VV, AR and IG (see Table 3-1). In general C_{jk} takes on three possible values: $extreme_1$, neutral and $extreme_2$ (e.g.: active, neutral and reflective for category AR). This approach allows us to consider the assessment step by step and therefore to reason between steps about which dimension would be the most useful one to compute next. This design also ensures that only the LS dimensions that each teacher is interested in are assessed.

3.6.3 Attributes of the System

As seen in the Learning Styles section, the learning styles have been broken down into categories. We are thus dealing with the following attributes:

- The answers to the questions
- The categories in each dimension
- The learning style dimensions

In addition, we are making inferences about four dimensions of learning styles. For each of these dimensions, the questionnaire asks about a criterion relevant to a style category. The system keeps asking questions, updating its degree of uncertainty until a desired threshold probability is reached, the system runs out of questions, or the student decides to stop answering questions.

3.6.4 Nature of the Questions

The questionnaire intentionally inquires about activities and preferences that have no relation to the classroom environment. The purpose is to make it difficult for students to infer the intent of the questions. The first assumption we are making is that the student does not know his or her learning style. Since the questionnaire is administered at first

online encounter, the nature of the questions avoids the case where the student gives the answer that he or she expects to give a good first impression about him or her. In the case of theoretical classes, the student may think that having a sensing style (rather than an intuitive one) gives a better impression about his or capabilities and may subsequently opt for answers that reflect that. Due to that reasoning, the questions relate to common activities that happen outside the classroom, with the intent of having the students answer the questions honestly.

Since there is no obvious direct relationship between questions and categories of the LS they are trying to assess, we attach a probability to the student's answer being 'yes,' 'no' or 'neutral,' given that the student belongs to a dimension's category. This probability is called the *conditional likelihood*. The notation of conditional likelihood is abbreviated as $p[Answer|Category]$ (e.g.: $p[Yes|Visual]$).

The questions that are administered to the candidate are stored in a database that is accessed by the inference engine. These questions come from two sources: ones we wrote, and others adapted from available learning style questionnaires. The latter were borrowed from different sources that have previously implemented self-assessment questionnaires. The sources that we have borrowed the questions from are the following: The Soloman and Felder Index of Learning Styles [37], the Fleming and Donwell VARK questionnaire [39], and modified questions from the Jewler and Gardner personal style inventory questionnaire [58].

3.6.5 Categorization of the Learning Style Dimensions

As mentioned previously, the categories of any given LS dimension are defined as: *extreme₁*, *neutral*, *extreme₂*. As for the answers to the questions, the traditional

Yes/No approach would only allow us to reach conclusions such as “Janet is intuitive” with 80% confidence. By letting the student answer Yes (Y), Neutral (M) and No (N), we can also show that the student does not really fit into any of the two extremes. The neutral response can be thought of as ‘maybe’ or ‘mixed’ (M).

We assessed the probabilities of the students responding Y, M or N if they belonged to an LS category. Let us consider the example of the visual/verbal LS categories. For these categories, the ‘format’ of the material presented is used to infer whether a subject is visual or verbal. In the knowledge base, multiple questions are relevant to this criterion. We assessed the probabilities of the candidate answering Y, M or N to these questions in the cases where he or she is a visual (C_{1k}), neutral (C_{2k}) or verbal (C_{3k}) individual. Therefore, we assigned nine likelihood estimates to each question relevant to category C_{jk} : $p[Y|C_{1k}]$, $p[M|C_{1k}]$, $p[N|C_{1k}]$, $p[Y|C_{2k}]$, $p[M|C_{2k}]$, $p[N|C_{2k}]$, $p[Y|C_{3k}]$, $p[M|C_{3k}]$ and $p[N|C_{3k}]$. These estimates relied on the literature of learning styles, the available LS questionnaires and our own judgment. Forty questions (each having on average three sub-questions) and their respective nine estimates of conditional likelihood are incorporated in the system’s knowledge base, totaling approximately a thousand values of conditional likelihood.

Table 3-3 provides an example of likelihood $p[A_{ir}|C_{jk}]$ for the question: “*You are going to cook something as a special treat for your family. Would you thumb through the cookbook looking for ideas from the pictures?*”

Table 3-3: Conditional Likelihood for Question Q₁

$A_{ir} \backslash C_{jk}$	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.9	0.25	0.1
Neutral	0.05	0.5	0.3
No	0.05	0.25	0.6

3.7 Assessment of Learning Style

The underlying computational methods are discussed in detail in this section. The implementation of the expert system updating functionality is detailed in section 3.7.2. The idea of a dynamic questionnaire was introduced in section 3.1. Here, the details of choosing the next question and administering it are discussed. We also introduce the notion of maximum value of information that will be the main driver of the order in which questions will be administered.

3.7.1 Adaptive Questionnaire Requirements

We set three major requirements for the system:

- Because of variability in any subject's answers to the questions and the relative relevance between questions and categories, we need several questions in order to assess the probability of each category along an LS dimension.

- At the same time, the answer to a single question should provide information about several dimensions. This is achieved by having sub-questions relating to a given situation, with each asking about an LS dimension.
- In order to optimize the interaction between the student and the system, the questionnaire has to be *adaptive* and *dynamic*. At each step, it assesses the current information gathered about the student and then determines the best question to ask next. The best question is the one that will provide the maximum value of information. This will be explained in further detail in sections 3.7.3 and 3.7.3.1.

The questionnaire is administered to assess the student's membership in one of the three categories ($extreme_1$, $neutral$, $extreme_2$) of each learning style dimension. We can think of these classifiers as bins, and the task is to estimate the probability that any given student belongs to each bin.

3.7.2 Bayesian Updating

We start with the assumption that the user belongs to each category with equal prior probability: $p[C_{1k}] = p[C_{2k}] = p[C_{3k}] = 1/3$. We will consider a question Q_1 in order to illustrate this. Let us consider a student's initial model at first encounter being the following: $p[Visual] = p[Neutral] = p[Verbal] = 1/3$.

Each set of questions starts by laying down the context. The setting is followed by three or four multiple-choice sub-questions relevant to the situation, with each sub-question inquiring about a learning style dimension. The context for the first question, Q_1 , is: "*You are going to cook something as a special treat for your family.*" This

following sub-question gathers information on the visual/verbal preferences of the student.

Q_1 : *Would you thumb through the cookbook looking for ideas from the pictures?*

Say the student's answer A_{1r} is Y (r represents the response). The objective is to calculate the probability that the student belongs to a learning style category given that he or she answered A_{ir} (Yes, in this case) to Q_i . We update the probabilities $p[C_{jk}|A_{ir}]$ using Bayes' rule:

$$p[C_{jk} | A_{ir}] = \frac{p[C_{jk}] \times p[A_{ir} | C_{jk}]}{p[A_{ir}]}$$

In this example, we get the following results (see Table 3-4):

- $p[C_{1k}|A_{1r} = Y] = 0.72$
- $p[C_{2k}|A_{1r} = Y] = 0.2$
- $p[C_{3k}|A_{1r} = Y] = 0.08$

This computation is based on knowing that:

- $p[C_{1k}] = p[C_{2k}] = p[C_{3k}] = 1/3$
- $p[A_{1r}|C_{1k}]$, $p[A_{1r}|C_{2k}]$ and $p[A_{1r}|C_{3k}]$ are known likelihood probabilities (see Table 3-3).
- $p[A_{1r}] = \sum_{j=1}^{j=3} p[A_{1r} | C_{jk}] \times p[C_{jk}]$

The change that we have done is to update $p[C_{1k}]$, $p[C_{2k}]$ and $p[C_{3k}]$ from $[1/3, 1/3, 1/3]$ to $[0.72, 0.2, 0.08]$. The maximum of these values is then compared to the desired threshold. In our experiments, the threshold τ was set to 0.8 (i.e. the questions on any given LS dimension continue until one of the posterior probabilities is greater than or

equal to 0.8). Since the desired value was not reached, additional questions are administered until either the threshold is met or the system runs out of questions. For the following question, the new values of $p[C_{1k}]$, $p[C_{2k}]$ and $p[C_{3k}]$ will be the new priors. Notice that by changing the threshold value, we can change the expected length of the questionnaire.

3.7.3 Question Sequence and Value of Information

One of the advantages of having a Bayesian updating system is the ability to assess the positioning of the student probabilistic model after each step. In the process of deciding which question to ask, the question sequence leads to varying lengths of the questionnaire. Our aim is to reach the desired threshold in the positioning of the student's LS categories, while keeping the questionnaire short and efficient.

Since each question has multiple sub-questions, it will give us information about multiple dimensions. Therefore, we choose at each step the question that has the maximum *expected value of information (EVI)*. In order to explain EVI, some concepts have to be introduced first in section 3.7.3.1.

3.7.3.1 Formulation of Information Value

In order to choose the question with the highest expected value of information, let us consider the simple case of deciding between two questions, with each having only one sub-question. We start by calculating the probabilities in the *perfect information* matrix of each sub-question. The *Perfect Information* matrix reflects the ideal situation of getting an answer from a question. In other words, if the student answers N to a question, the perfect information matrix gives the value of absolute certainty that the student has a

Verbal style (see Table 3-5). This matrix is based on the *posterior probabilities* matrix (see Table 3-4). For every question Q_i relating to dimension k , the matrix entries are defined by:

$$Posterior(r, j) = \frac{p[C_{jk}] \times p[A_{ir} | C_{jk}]}{p[A_{ir}]}$$

Table 3-4: Posterior Probability Matrix for Q_1

$P[C_{jk} A_{ir}]$	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.7200	0.200	0.0800
Neutral	0.0588	0.5882	0.3529
No	0.0556	0.2788	0.6667

Along each row of the *posterior probabilities* matrix, we find the maximum entry, and we set the corresponding entry to the perfect probability. The other entries are set to 0 (see Table 3-5). In order to account for equal maxima, we define the set:

$$S_r = \left\{ j \mid Posterior(r, j) = \max_d Posterior(r, d) \right\}$$

with d being a dummy variable.

$$\forall r, j \quad Perfect(r, j) = \begin{cases} 1/|S_r|; & j \in S_r \\ 0; & otherwise \end{cases}$$

With $|S_r|$ being the cardinality of the set S_r .

The case where two posterior probabilities are equal is accounted for as a special case. In order to avoid having two perfect probabilities of 1, which will violate the total probability theorem, one of the maximal equal probabilities is randomly set to 1.

Table 3-5: Perfect Information Matrix for Q1

	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	1	0	0
Neutral	0	1	0
No	0	0	1

The *penalty* matrix is defined as the measure of how far off the posterior probabilities are from perfect information. The *penalty* matrix is obtained by taking the absolute value of the difference between the posterior probabilities and the *Perfect Information* matrix entries. The closer a posterior probability is to the perfect information, the less the question is penalized (see Table 3-6).

$$\forall r, j \text{ Penalty } (r, j) = | \text{Posterior } (r, j) - \text{Perfect } (r, j) |$$

Table 3-6: Penalty Matrix for Q₁

	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.2800	0.200	0.0800
Neutral	0.0588	0.4118	0.3529
No	0.0556	0.2778	0.3333

The final step is to calculate the EVI of each question. The *Perfect Information* of the 3x3 matrix is the sum of *Perfect*(*r, j*) entries. The *penalty* is the sum of the *penalty* matrix entries.

The EVI of each question is the difference between the *Perfect Information* and the *penalty*. The question with highest EVI will be the next question to administer.

$$EVI(Q_i) = \sum_{r,j} Perfect(r, j) - \sum_{r,j} Penalty(r, j)$$

$$EVI(Q_1) = 3 - 2.0502 = \mathbf{0.9498}$$

3.7.3.2 Question Sequence

In order to illustrate the question sequence that this system will generate, let us consider a second question, Q_2 :

Q2: To decide whether you want to watch a movie, would you read the reviews?

Setting
Sub-question

The likelihood estimates (Table 3-7), the posterior probabilities (Table 3-8), the perfect information matrix (Table 3-9) and the penalty matrix (Table 3-10) for Q_2 follow.

Table 3-7: Likelihood Matrix $p[A_{ir}|C_{jk}]$ For Q_2

$A_{ir} \backslash C_{jk}$	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.1	0.1	0.8
Neutral	0.2	0.8	0.1
No	0.7	0.1	0.1

Table 3-8: Posterior Probability Matrix for Q_2

$p[C_{jk} A_{ir}]$	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.1000	0.1000	0.8000
Neutral	0.1818	0.7273	0.0909
No	0.7778	0.1111	0.1111

Table 3-9: Perfect Information Matrix for Q₂

	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0	0	1
Neutral	0	1	0
No	1	0	0

Table 3-10: Penalty Matrix for Q₂

	<i>Visual</i>	<i>Neutral</i>	<i>Verbal</i>
Yes	0.1000	0.1000	0.2000
Neutral	0.1818	0.2727	0.0909
No	0.2222	0.1111	0.1111

After calculating the question's *perfect information* and *penalty* matrices, the EVI of the question results in: $EVI(Q_2) = 3 - 1.3899 = \mathbf{1.6101}$

Comparing Q₁ and Q₂, Q₂ promises more *value of information* about the student's current model and will therefore be the next question to be asked. This *information value* is relevant to one sub-question. We therefore sum the values of the sub-questions, and the question with the highest total *EVI* will be the next one to be asked from the pool.

In order to avoid unnecessary questions, any sub-questions relevant to an LS dimension that has reached the threshold for one of its categories will be removed from that question. This method would improve the questionnaire in matters of conciseness, avoiding unnecessary time and effort expenditure for the student.

3.7.4 Determination of the Learning Styles

The system stops asking questions about an LS dimension by either reaching the threshold or running out of questions about that dimension. In both cases, it returns the category with highest probability. A large enough pool of questions can help avoid the latter case, but will not completely circumvent it for some respondents. In other words, the system makes its best effort to reach the desired threshold.

3.8 Results

We ran the system internally at the MIT Center for Educational Computing Initiatives on 18 subjects (students and researchers), and we obtained the results in Table 3-11. In addition to the LS dimensions that we presented previously and that will be used in the MAS system, we were assessing three more LS dimensions: Sequential/Global, Aural/Read-Write and Kinesthetic/Passive. It is worth noting that most individuals did not fit into extremes in general, except for the Sequential/Global and Individual/Group categories where most of the subjects were categorized as sequential and individual-oriented learners. This comes as no surprise, since the pool of people comes mainly from an academic environment.

Table 3-11: Results of Learning Style Assessment

	<i>Extreme₁</i>	<i>Neutral</i>	<i>Extreme₂</i>
Sensing Intuitive	9%	82%	9%
Visual Verbal	0%	100%	0%
Active Reflective	0%	73%	27%
Individual Group	64%	36%	0%
Sequential Global	55%	36%	9%
Aural Read-Write	45%	55%	0%
Kinesthetic Passive	27%	64%	9%

The subjects answered on average 17.5 context questions with a standard deviation of 5.2. Within these context questions, the number of sub-questions that they answered was 50.4 on average, with a 16.6 standard deviation. The desired threshold τ was set at 0.8 for all dimensions. All dimensions reached a confidence level above this threshold for all subjects. As a comparison measure, the standardized ‘Building Excellence Inventory’ [107] assesses learning styles by using 118 multiple-choice questions according to the Dunn & Dunn learning styles model (D&D), and the Hermann Brain Dominance Instrument (HBDI) consists of 120 questions. In both cases, the number of expected answers is cut by more than half by our adaptive Bayesian approach.

One advantage of the expert assessor is the guarantee that users who are more consistent in their answers will get a shorter questionnaire. This will avoid them having to spend a long time answering the questionnaire. The average time spent to answer the D&D questionnaire was 25 minutes, and the HBDI questionnaire averaged 30 minutes.

With our expert LS Assessor, subjects spent an average time of 7 minutes answering the questions. This duration is promising in terms of students answering the questionnaire diligently. Another advantage is the ease of tuning the threshold to the desired level. Advantages of threshold setting extend in both directions: A lower threshold will result in a shorter questionnaire length, and a higher threshold will result in a greater confidence level in the LS profile of a subject.

The mathematical rigor of the system makes it superior to popular and commercially available ones that consist of a fixed number of questions and that compute the average of all the questionnaire answers. The popular approach is easy to implement, but has the disadvantage of lengthy questionnaires. In our system, the addition of the conditional likelihood estimates of the student's answers given their membership in a LS category is reasonable. By casting doubt about each answer's relation to the LS, we are holding to the assumption that a question does not absolutely indicate that the student's style fits into a certain category.

The Bayesian updating system provides a way of efficiently assessing the student's LS. The EVI attached to each question makes the assessment as short as possible, while putting the best effort to reach the desired threshold value. After administering the questionnaire, we have an initial student model. The MAS is now able to start suggesting course content relevant to the student's cognitive style. This model will be updated according to the website usage. The student's actions (saving a document, clicking on a link, spending time on a web page...) will be used to refine the initial student model and to adapt to a varying learning style.

Chapter 4 – Agreement of Multi-Agent System

Recommendations

There is a need to teach online students according to their preferred learning style (LS). Dede [32] notes that classroom settings narrow the range of students' learning styles (LSs) in comparison to teaching where multiple activities are available, in reference to web-based education. However, in a media-rich online environment, searching for the right media formats and activities can be time-consuming and can potentially lead the students to spend ineffective time finding their preferred activity.

Our goal is to provide the students with a more personalized ordered list of search results when they wish to learn about a course topic or keyword. By taking the students' LS into account, the presentation of the course material can be personalized to match the students' preferences. This chapter presents a new mathematical foundation to increase the relevance of search results based on the student's LS profile. Bollen [15] has shown that *adaptive link ordering* (ordering search matches according to student model) improves selection time and reduces cognitive overhead.

This chapter presents some previous efforts in adaptive link ordering in section 4.1. It continues with an introduction to the multi-agent system (MAS) that models the

user's preferred learning style in section 4.2. Once the agents have a student model, their main contribution comes at the time when the student searches for a new topic or enters a query about a keyword that they encountered. In that case, each agent has a probabilistic student profile (or model) along one learning style dimension. Therefore, each agent will make a recommendation of the types of media that match the student's model. These recommendations are made as a ranked list of media types ranging from most to least preferred. The methodology for agents' ranking of educational media is presented in section 4.3. Since the MAS consists of four expert agents that make these recommendations, the agents will frequently disagree. Section 4.4 introduces the need to resolve the disagreement among the agents' recommendations. Section 4.5 discusses various voting approaches to resolve these conflicts. The best-fitted voting algorithms will then be applied to the problem at hand in section 4.6. The concluding section discusses the usage of the adaptive link ordering in the search results phase and presents the advantages of the MAS approach, in particular the way it matches the educational media to the student.

4.1 Related Work

Paredes et al. [95] presented a system for content sequencing according to LS profiling using the Felder-Silverman LS model [37]. They explain how to combine two LS dimensions (sequential/global and sensing/intuitive) to sequence the educational media. They mention that incorporation of additional LS dimensions in their approach can be problematic and do not solve it.

Carver et al. [26] described the incorporation of LSs into hypermedia courseware. The aggregation of multiple LS dimensions to decide on which media to present to the

student is done heuristically, and it is not clear how they calculate it. They chose the Felder-Silverman model to apply it to the course media without justification for this particular choice. They simply state, “The authors believe that the Felder Model is most appropriate for hypermedia courseware.” There was no formal quantitative assessment of the usefulness of LSs incorporation. They assert, “Instructors teaching CS383 have noticed a dramatic change in the depth of student knowledge,” and “it appears that the best students benefit the most from hypermedia courseware and the worst students benefit the least.”

Wolf [124] uses the Dunn and Dunn model without giving a justification for this selection. He worked on learning style-based e-learning and mentions that “it still remains unclear what aspects of a learning style profile are worth modeling and which is the most effective approach for a particular style”.

4.2 Multi-Agent System Functionality

In our work, the multi-agent tutoring system’s responsibility is to decide on the most suitable educational media to present to the student, basing the recommendations on the subject’s LS and past actions. Our definition of agent is an entity or program that is independent, proactive, task-oriented, knowledgeable about its task, information-acquiring, and aware of changes in its environment [57], [125].

Every agent is responsible for observing and updating a LS dimension. The agent builds a profile of the student along that dimension and recommends a ranked list of educational media to the student. The student’s LS is available to the MAS at each point in time by updating the student profile after each student action. For example, if the user

clicks on a problem set hyperlink, this is taken as an indicator that there is interest in more active and concrete activities. The expert agents in perception style (Sensing/Intuitive) and processing style (Active/Reflective) recalculate their respective probabilistic estimate of the student's LS profile by using Bayesian updating. The LS profile consists of several dimensions that are described below. When the student searches for a new topic, different tutoring agents follow different aspects of the student's LS. When they disagree on the teaching media that they are going to present to the student, an aggregator agent requires them to cast their votes and rank the educational media. The aggregator then treats the agents' votes as inputs to a multi-criterion, multi-preference ranking problem and finds the ranking that reconciles all the votes.

The estimation of the student's LS profile was discussed in Chapter 3. Briefly, a knowledge-based system starts by administering a dynamic questionnaire to the student to assess his or her LS by asking the minimal number of questions and creates a satisfactory initial student profile based on the answers. The result is a probabilistic assessment of the student for each LS dimension (e.g.: The active/reflective LS style profile of student U is $p[\textit{reflective}] = 0.7, p[\textit{active}] = 0.2, p[\textit{neutral}] = 0.1$).

After this initial assessment, the MAS builds on the initial assessment by recording the student's online interest in teaching media as measured by time spent, student's action to save a document or follow a link, and frequency of accessing certain media types. This will be used by the individual tutoring agents to update their beliefs about the student's LS. There is no research evidence that proves that individuals' LSs are constant over time. Snow's [113] and Kozma's [71] findings indicate that learning

styles can vary with different tasks or different learning content. Therefore, the agents constantly monitor the evolving student's interests.

4.3 Individual Agents' Ranking of Educational Media

In our context, we are dealing with the subtle issue of LS of the student. The information that we have is the probability of the student having a preference (e.g.: *probability [student having a visual style]*). The agents' task is to assess what teaching media fit the student's profile best. Since a tutoring agent does not have a clear preference for matching a profile to the teaching media, we have an extra complication in comparison to classical decision-making. Let us consider as an example a student "*U*" with the following probabilistic profile along three learning style dimensions:

- Visual-Verbal dimension: $p[visual] = 0.8, p[verbal] = p[neutral] = 0.1$
- Sensing-Intuitive dimension: $p[sensing] = 0.9, p[intuitive] = 0.06, p[neutral] = 0.04$
- Active-Reflective dimension: $p[active] = 0.2, p[reflective] = 0.7, p[neutral] = 0.1$
- Individual-Group dimension: $p[individual] = 0.3, p[group] = 0.5, p[neutral] = 0.2$

The first step is to rank the teaching media according to each agent's domain (which can be thought of as a criterion for choosing among alternatives). In Figure 4-1, we describe the teaching media in terms of the LS that they are closest to (e.g.: simulations are more 'visual' than problem sets are). When two media are assessed to be equally ranked, then the notation is $medium_1-medium_2$. For example, on the Visual-

Verbal dimension, the notation T-F denotes that the textbook and the FAQ's are equally preferred.

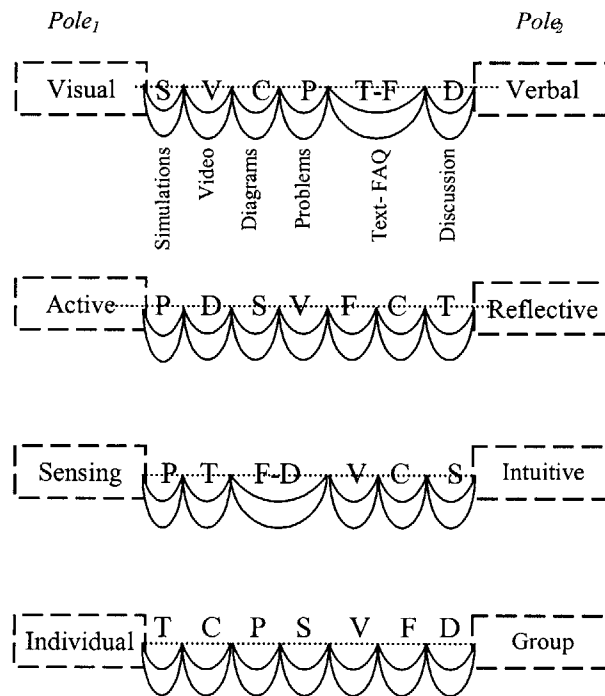


Figure 4-1: Graphical Distribution of Teaching Media Over Learning Style

Dimensions

The agents model their preferences for media in a probabilistic fashion. For example, agent VV (Visual-Verbal) estimates that the preference probability of a visual student to watch a video segment is 0.25 (see Table 4-1). Given the seven media types that PIVoT contains, a probability distribution is assigned to each category (visual, verbal, and neutral). Table 4-1 shows the matrix of conditional likelihood of the media given the LS dimension's categories. We developed the conditional likelihood values for

the remaining LS dimensions, based on the literature search and our own judgment. These can be found in Appendix A.

Table 4-1: Distribution of Conditional Likelihood of Media Over Types

$p[\text{medium} \text{type}]$	Video	Simulation	Problem	Text	Discuss	FAQ	Diagram
Visual	0.25	0.3	0.1	0.06	0.03	0.06	0.2
Verbal	0.06	0.04	0.15	0.2	0.25	0.2	0.1
Neutral	0.08	0.04	0.25	0.2	0.03	0.2	0.2

In order to calculate agent VV's ranking, we use the total probability theorem.

Theorem 4-1:

If $\{ B_n : n = 1, 2, 3, \dots \}$ is a finite or countably infinite partition of a probability space and each set B_n is measurable, then for any event A we have:

$$p[A] = \sum_n p[A | B_n] \times p[B_n]$$

In the case of the media, agent VV's preferences will be calculated as:

$$\begin{aligned} p[\text{video}] &= p[\text{video} | \text{visual}] \times p[\text{visual}] \\ &+ p[\text{video} | \text{verbal}] \times p[\text{verbal}] \\ &+ p[\text{video} | \text{neutral}] \times p[\text{neutral}] \end{aligned}$$

$$p[\text{video}] = 0.25 \times 0.8 + 0.06 \times 0.1 + 0.08 \times 0.1 = 0.214$$

Similarly, we get the following values for the remaining educational media:

$$\begin{aligned} p[\text{sim}] &= 0.248, p[\text{prob}] = 0.12, p[\text{text}] = 0.07, p[\text{discuss}] = 0.052, \\ p[\text{FAQ}] &= 0.088, \text{ and } p[\text{diagrams}] = 0.19 \end{aligned}$$

After sorting these values, agent VV will have the following ranked preference: (1) Simulation, (2) Video, (3) Diagrams, (4) Problems, (5) FAQ, (6) Text and (7) Discussion.

Agents SI (Sensing-Intuitive), AR (Active-Reflective), IG (Individual-Group) will have their own preferences by the same calculations seen above. It is easy to see that ties between two media can occur; this is acceptable for the current purposes. After gathering all the agents' preferences, the next task is to provide the user with a ranked preference that aggregates the various recommendations in a way consistent with the user's LS profile.

We will now consider different voting and ranking methods in order to gather the different agents' recommendations, discuss their advantages and drawbacks, and conclude with the best-fit methods for multi-agents/multi-options systems.

4.4 Multi-Agent System and Conflict Resolution

Many publications describe multi-agent systems for pedagogical help of online learners. Most of the work was designed to aid and guide the students in procedural tasks such as medical diagnosis (e.g. ADELE [60]) and problem solving (e.g. Herman the Bug [73]). We are dealing with the general situation of a student browsing through a course website and learning new concepts or revisiting material.

We chose multi-agent systems to solve this problem for various reasons:

- Agents have desirable properties: independence, pro-activity, knowledge possession, capability of acquiring information, task-orientation, and awareness of changes in their environment.

- Agents can be made experts in their field: each agent is an expert in a LS dimension and can observe the student's relevant behavior, and can recommend tutoring media according to the student's profile in their LS dimension context.
- Agents can interact with each other and behave in a "society" where they recognize other agents' needs to communicate with respective priorities.
- Although the previous reasons can all be incorporated into simple programs and objects, 'agents' are a practical way of thinking about a distributed problem-solving task.

Each agent has a profile of the student and will suggest the different media that match his or her needs. Nevertheless, there is a possibility of conflicting suggestions between the agents. Therefore, there is a need to reconcile the suggestions of the agents by using an aggregator agent.

We are assuming the tutoring agents' recommendations equally important. In addition, the agents have no knowledge of the other agents' recommendations. This approach has the advantage of adapting to new models or model dimensions that may be added in the future, without affecting the existing system. The drawback is that knowledge is not shared, and individual agents do not capture interdependent LS dimensions. With our approach, as an example, a tendency of active subjects to be visual as well will be reflected by having changes in the student's profile that agents *AR* and *VV* are monitoring, without their mutual knowledge of these changes.

In order to reconcile the agents' suggestions, an aggregator agent will let them rank the educational media that, from their perspective, are most suitable to a student's preferences at a point in time. The agents rank the educational media by first, second, third preference, etc. By taking these preferences into account, the perils of plurality voting are avoided. Plurality voting takes the first preference and ignores all subsequent preferences. As Saari [108] puts it in describing three way elections, "Plurality vote is the only procedure that will elect someone who's despised by almost two thirds of the voters." By taking second and subsequent preferences, we make sure that all the agents' votes and preferences are accounted for.

First we will introduce the voting and ranking schemes that were previously devised; we will proceed with the Borda, Kohler and Arrow-Raynaud's methods of multi-criterion ranking application. We will show how each tutoring agent views the educational media from a different perspective, and how we can apply the ranking methods in order to reach an agreement on the best alternatives that fit a student profile. We will then discuss the limitations of the different methods, and explain which methods fit our needs best.

4.5 Ranking and Voting

There are two ways to proceed about ranking. Cardinal methods, such as the multiple attribute utility theory (MAUT) [66] and the analytic hierarchy process (AHP) [109], require expert assessors to express their degree of preference for an alternative over another for each criterion. Ordinal methods require only that the rank order of the alternatives be known for each criterion [72]. We opt here to rely on ordinal methods.

Many ordinal ranking methods have been devised. Saari [108] discusses the pitfalls of plurality voting (see section 4.4) and the Borda count in the context of politics and voting. Borda proposed a system of ranking by giving each candidate points corresponding to their rank, by having voters give one point to the least preferred candidate, the next candidate two points, and so on until they reached their most preferred candidate to whom they would give n points. Some functionally equivalent versions of the Borda method score the alternatives from 0 to $n-1$.

The Marquis de Condorcet [27] wrote one of the seminal works on finding a successful vote winner, though his text was very complex and the ideas were not clearly explained. Therefore, Young [127] clearly explained Condorcet's theory; the Condorcet winner is the alternative that wins a pair-wise election vote against any other alternative. Condorcet's criteria generate the Condorcet order (or a complete or preference order) as: a_1, a_2, \dots, a_n , and a_i beats a_j in a pair wise election if $i < j$. This will be a ranking of alternatives that we will try to attain by aggregating the votes of multiple agents. Young also points out the reasons that make Condorcet's method a superior one to Borda's: "...Condorcet's method gives the ranking of the outcomes that is most likely to be correct."

Arrow ([5], [6]) investigated the possibility of a method of voting which would embody the requirements of a voting system:

- Universality: The voting method should provide a complete ranking of all candidates from any set of individual preference ballots.
- Monotonicity: If one set of preference ballots leads to an overall ranking of candidate A above candidate B, and if some preference ballots are changed in

such a way that the only candidate that has a higher ranking on any preference ballots is A, then the voting method should still rank A above B.

- Independence of irrelevant alternatives: If one set of preference ballots leads to a ranking of candidate A above candidate B and if some preference ballots are changed without changing the relative rank of A and B, then the method should still rank A above B.
- Citizen sovereignty: Every possible ranking of candidates can be achieved from some set of individual preference ballots.
- Non-dictatorship: There should not be one specific voter whose preference ballot is always adopted.

Arrow concluded that it is not possible to have a voting method with all of these properties. Later, he published with Raynaud [6] the different methods (including his) that try to find a complete (Condorcet) order of preferences. Lansdowne [72] compares the properties that can be attained by ordinal ranking methods that were devised by Borda [16], Bernardo [13], Cook and Seiford [29], Kohler [68], and Arrow and Raynaud [6]. The first three methods guarantee neither a Condorcet order nor a Condorcet winner. Kohler's algorithms ensure both (as long as a criterion does not equally prefer two alternatives), and Arrow-Raynaud's ranks the alternatives according to a Condorcet order, but does not necessarily place the Condorcet winner in first place. Nevertheless, Kohler's [68] limitation is that it does not fulfill the increasing sequential independence principle that requires the relative rank order of the best r out of n alternatives to be a function of only those alternatives, where r is any integer from 1 to n .

Before we proceed, let us define some notions:

- A ‘complete order’ (or ranking) means that an agent has to prefer one of every two alternatives, and therefore the ranking does not have ties in it.
- ‘Prudent order’, introduced by Arrow and Raynaud, is attained when the ranking is a preference order (complete order) and it contains no cycles.
- ‘Cycles’ occur when an order of the sort $a_1 \succ a_2, a_2 \succ a_3 \dots a_{n-1} \succ a_n, a_n \succ a_1$ places an alternative as least and most preferred (\succ denotes “is preferred to”). In a multi-criteria multi-voter ranked preferences voting, cycles occur approximately 10% of the time [27]. A demonstration of a cycle is represented in section 4.5.1

4.5.1 Cycles

Table 4-2 illustrates a voting example with seven voters on four options. The voters rank their preferences from one to four.

Table 4-2: Cycles Illustration

	Voter 1	Voter 2	Voter 3	Voter 4	Voter 5	Voter 6	Voter 7
Rank 1	A	B	B	C	D	D	D
Rank 2	B	C	C	B	C	C	A
Rank 3	D	A	D	D	B	A	C
Rank 4	C	D	A	A	A	B	B

The next step is to consider each pair of options apart and observe the winner among these two. This is illustrated in Table 4-3 where the entries $a(i,j)$ represent the number of voters that prefer the i^{th} row option to the j^{th} column option.

Table 4-3: Voters' Preferences

	A	B	C	D	Comparisons
A	-	3	2	2	
B	4	-	3	4	B>D
C	5	4	-	3	C>B
D	5	3	4	-	D>C

B has an additional voter that prefers it to D, C has one additional voter that prefers it to B, and D wins against C. The cycle is illustrated in Figure 4-2, with the arrows pointing from the winner to the loser.

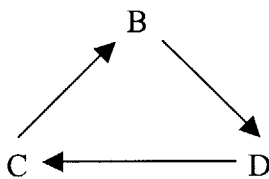


Figure 4-2: Cycle Illustration

4.6 Ranking Methods

We examine Borda [16], Kohler [68] and Arrow-Raynaud's [6] methods for the demonstration of important properties: Borda's method (like Bernardo's and Cook-Seiford's) guarantees neither the Condorcet winner nor the Condorcet order. Nor does it satisfy the increasing or decreasing sequential independence principles. Nevertheless, we applied it for its popularity in voting events and in order to compare its results with the more rigorous (Kohler and Arrow-Raynaud) methods.

4.6.1 Borda's Method

By using the Borda voting method, voters rank the candidates as first, second, third, etc. The Borda [16] voting procedure gives the highest ranking preference $n-1$ points, the second preference $n-2$ points, etc. The points from each voter are summed to determine the winner.

The Borda election method is one of the better-known methods for tallying ranked ballots. It is used in various scientific and technical applications such as handwriting recognition [44], where the votes come from unbiased sensors or systems rather than people. The Borda system has received a disproportionate share of attention in the popular press.

Let us consider U 's profile in three learning style dimensions to illustrate the basis idea behind multi-preference voting methods. Subject U has the following LS profile:

- $p[visual] = 0.8, p[verbal] = 0.1$ and $p[neutral] = 0.1$
- $p[sensing] = 0.9, p[intuitive] = 0.06$ and $p[neutral] = 0.04$
- $p[reflective] = 0.7, p[active] = 0.2$ and $p[neutral] = 0.1$
- $p[individual] = 0.3, p[group] = 0.5$ and $p[neutral] = 0.2$

By applying the total probability theorem (see Equation 4.1), the agents can express their preferences towards the educational media given the current student's profile. Agent VV will give "simulations" 6 points; "videos" gets 5 points, and "diagrams" 4 points. We proceed in this fashion to rank all media for the three agents. The aggregator agent will then sum up the score that was assigned to each teaching medium. This procedure will aggregate the tutoring agents' recommendations and produce one final ranked recommendation. The result obtained does not guarantee having a single winner, but this does not matter in our context. There is no drawback to having two teaching media being equally preferred by the student. Nevertheless, Borda's method does not ensure that the final ranking is the Condorcet order. It is a rather popular method for the ease of computing the results, and for being an intuitive approach. Table 4-4 shows the resulting media ranking for subject *U*:

Table 4-4: Agent's Assessment of Subject *U*'s Preferences Using the Borda Count

	Agent VV	Agent SI	Agent AR	Agent IV	<i>Total Score</i>	<i>Rank</i>
Video (V)	5	4	3	3	<i>15</i>	<i>2</i>
Text (T)	1	0	6	5	<i>12</i>	<i>4</i>
Problem (P)	3	6	5	6	<i>21</i>	<i>1</i>
Simulation (S)	6	5	1	2	<i>14</i>	<i>3</i>
Chart/Diagram (C)	4	1	4	1	<i>10</i>	<i>5</i>
FAQ (F)	2	3	2	1	<i>8</i>	<i>6</i>
Discussion (D)	0	2	0	0	<i>2</i>	<i>7</i>

Borda's method results in the following positioning of media for subject U : P, V, S, T, C, F, and the least preferred being D.

4.6.2 Kohler and Arrow-Raynaud

In order to discuss the Kohler and Arrow-Raynaud primal and dual algorithms, we first introduce the notion of outranking matrix, which they operate on. This $[n \times n]$ matrix denotes the votes that each medium got in comparison with other media. We define a_{ij} to be the number of agents ranking the i^{th} medium before the j^{th} medium.

The methods developed by Kohler and Arrow-Raynaud are based on the diversity axiom that states that *each criterion should be a total order on a finite set of n alternatives*. This is not the case for a tutoring agent, which sometimes equally prefers two educational media. This is translated in the outranking matrix by not having constant sums of a_{ij} and a_{ji} . An example shows that there is one agent that equally prefers “text” and “FAQ” (bolded in Table 4-5). The example in Table 4-5 shows the outranking matrix for subject U .

Table 4-5: Outranking Matrix

	V	S	P	T	D	F	C
V	-	1	1	3	4	3	2
S	3	-	2	2	3	2	3
P	3	1	-	3	4	3	2
T	1	1	1	-	2	1	1
D	0	0	0	1	-	0	2
F	0	2	0	1	3	-	2
C	1	1	1	2	2	2	-

4.6.2.1 Kohler's Approach

Kohler's algorithms perform a local optimization at each step and determine the best alternative at hand.

4.6.2.1.1 Primal algorithm

Step p : Find the minimum along each row of the current matrix, and determine the maximum of these minima. If there are ties, choose the maximum arbitrarily. This is our best p^{th} alternative. Eliminate the row and column corresponding to that alternative, and proceed with the $(p+1)^{th}$ step if the matrix is not empty.

4.6.2.1.2 Dual algorithm

Step p : Find the maximum along each column of the current matrix, and determine the minimum of these maxima. If there are ties, choose the minimum arbitrarily. This is our best p^{th} alternative. Eliminate the row and column corresponding to that alternative, and proceed with the $(p+1)^{th}$ step if the matrix is not empty.

4.6.2.2 Arrow-Raynaud's Method

Lansdowne [72] formalizes the Arrow-Raynaud algorithms in the following two methods. These algorithms find the losing option first and finally find the winner.

4.6.2.2.1 Primal algorithm

Step p : Find the maximum along each row of the current matrix, and determine the minimum of these maxima. If there are ties, choose the minimum arbitrarily. This is our worst $(n-p)^{th}$ alternative. Eliminate the row and column corresponding to that alternative, and proceed with the $(p+1)^{th}$ step if the matrix is not empty.

4.6.2.2.2 Dual algorithm

Step p : Find the minimum along each column of the current matrix, and determine the maximum of these minima. If there are ties, choose the maximum arbitrarily. This is our worst $(n-p)^{th}$ alternative. Eliminate the row and column corresponding to that alternative, and proceed with the $(p+1)^{th}$ step if the matrix is not empty.

4.7 Discussion of the Methods

The aim of the two approaches is to provide a *prudent* final rank ordering (see section 4.5). This is guaranteed when there are no ties in agents' preferences. Nevertheless, a theorem (Arrow and Raynaud) that is relevant to the case of ties states the following:

Theorem 4-2:

Even if ties are present in the criteria rankings, even if the solution is not unique, Arrow-Raynaud's primal algorithm results in a preference order free of cycles, and Arrow-Raynaud's dual algorithm results in a preference order that is complete.

Essentially, the theorem states that the primal algorithm results in an order without cycles, and the dual algorithm delivers a preference order.

Our concern is not to have cycles, and therefore the dual algorithm is sufficient for our case when ties are present. A complete order is not necessary, since we can have a final partial order where the agents will suggest that two media are equally relevant to the student's interests. Similarly, Kohler's primal algorithm guarantees a no-cycle result that is sufficient for our purposes.

We ran the algorithms on multiple student profiles. The rankings returned by Arrow-Raynaud's primal and Kohler's dual algorithms were similar 80% of the time. The rankings returned by Arrow-Raynaud's dual and Kohler's primal algorithms were similar 93% of the time. Borda's method resulted in a ranking close to the four algorithms; but,

as we mentioned earlier on, this method does not provide us with a social order where the higher-ranking alternative beats all the lower ranking ones.

4.8 Multimedia Ranking Results

In PIVoT, when the user submits a query, the search engine will return results based on keyword and topic match, which act as the initial ranking criteria. Say we get results the results Text_1 (T402), Video_1 (V503) and Simulation_1 (S132) in this order. The search engine assigns a percentage value to the search results proportional to the media's relevance (see Figure 4-3). However, the agents' recommendation ranked text, videos and simulations as 4th, 3rd and 1st respectively for subject U (see section 4.7). The results are therefore passed through an LS filter that readjusts the rankings based on the agents' final recommendation. The LS filter consists of a multiplier inversely proportional to the media ranking. In other words, the most preferred learning medium, in this case “simulation”, gets multiplied by 1, leaving it unaffected. All subsequent preferences will be assigned decreasing multipliers that lie within the pre-specified initial range [0.4; 1].

The lower limit of the multiplier range changes according to the user's interest in the presented search results. When the student repeatedly clicks on poorly ranking search results, the range of multipliers is decreased, and vice versa. By changing the multiplier range, the learning styles' effect on search results is dynamically dampened or amplified in response to the media's relevance and the user's interest.

We set the lower bound (LB) to decrease if the student clicks on the first, second or third search result, stay constant if the student clicks on the fourth search result, and

increase if the student opts for higher order results. Mathematically, this is expressed with the following pseudo-code:

$LB = LB + (\text{search_result_rank} - 4) \times 0.1$

If $(LB < 0.4)$, $LB = 0.4$

If $(LB > 1)$, $LB = 1$

SeARCH

law

VIDEO

L11g Newton's Law of Universal Gravitation (0:14:27) CC 26%

Newton's law of universal gravitation is introduced, and the gravitational force falls off as one over the distance squared. In general the gravitational potential energy is zero at infinity, and its always negative as you move in towards a massive object from infinity, and is inversely proportional to the distance from the object. It is consistent with the small distance approximation near the Earth's surface that the gravitational potential energy is mgh.

SG Pucks on a Horizontal Frictionless Surface (0:02:07) 16%

Discussion of the $F=ma$ equation (Newton's second law), net force, and motion when the acceleration is zero (Newton's first law).

SG Pulling a Tandem Pair of Sleds (0:06:25) 16%

A father pulls his daughter in a sled on a frictionless surface; her sled is towing a second sled. A free body diagram is drawn for each sled, and the net force equations are found for the horizontal and vertical directions, which can be solved to determine the tension in each rope. The effective weight of the daughter is decreased because her father is pulling with a vertical component.

TEXTBOOK

Chapter 5: Dynamics--Newton's Laws 17%

Chapter 5: Summary 17%

Chapter 9.1: Newton's Law of Universal Gravitation 10%

Figure 4-3: PIVoT Search Results

4.9 Conclusion

This chapter presented the agents' tasks of suggesting their preferences for educational media presentation according to each agent's view of the student's profile. The aggregator agent subsequently has the agents cast their votes, and it performs calculations giving all the agents equal voting power, resulting in a ranked partial order of the educational media matching the student's current LS profile. The contributions of this approach are both educational and computational.

On the educational front, we have seen that previous similar systems adopted a specific LS model without justification for the rationale of their choice ([26], [93], [124]). We looked at the abundant literature about LSs and analyzed them methodically to select LS dimensions that lead to increased learning and improved learning experience.

On the computational front, we have devised a new mathematical foundation to increase the relevance of search results according to the student's multi-dimensional LS. We have seen that previous research on multi-dimensional LS incorporation was problematic [93] or done heuristically [95]. We have presented the different rigorous multi-criteria voting methods applied to a MAS. We found that by applying Kohler's primal or Arrow-Raynaud's dual algorithms to the ranking of teaching media by different agents, we achieve a resulting cycle-free unified ranking that will be the agents' collective recommendation.

Chapter 5 – Software Architecture

This chapter offers a more complete discussion of the design of both PIVoT and the multi-agent system. As discussed earlier, the expert system interacts with the student at first encounter in order to build an initial model of the student's learning style. The system saves the student's answers to the questionnaire, along with the inferences made about the student's learning style in a database that can be accessed by the multi-agent system. After the student answers the questions, he or she is able to browse through the course website and explore the various educational media and material. The multi-agent system observes all the student's actions and records them as indicators of a continuously evolving learning style. When the student searches for a new topic, the Multi-Agent System (MAS) will suggest the learning material that fits the current learning style model.

We will discuss the technological details of this approach and how the seamless system is able to assist the student with minimal direct interaction.

5.1 Intelligent Education and the Web

Devedzić [33] states, “Designing the architecture of an Intelligent Tutoring System (ITS) involves a large measure of art.” As such, this section does not pretend to offer a taxonomical view of ITS architectural design, but rather a collection of sample approaches to this important task.

With agents being a key metaphor in ITS design, the issue of single-agent versus multi-agent systems becomes important. Badjonski *et al.* [7] first proposed the use of multi-agent systems in ITS design. They noted that multi-agent systems facilitate easier development by using a “divide and conquer” approach to break down complex tasks into several simpler entities. Additionally, agent-oriented programming primitives are easy for programmers to use. Further, an MAS approach simplifies modification or expansion and enables the benefits of mobile agents and distributed computing [42].

Another common architectural theme in ITS design is the use of competing, specialized, differentiated agents to handle learner diversity ([44], [74], [106]). Lelouche [74] advocates using collections of agents with specialized functions that handle the issue of learner diversity. Alternatively, Heift and Nicholson [44] discuss the importance of “generality, modularity and efficiency” in ITSs for handling learning diversity. Roselle and Grandbastien [106] go further, proposing a platform for educators to design educational experiments and combine them into a single educational software package. These approaches are popular due to their obvious ability to reduce overall costs of ITS development by encouraging code reuse.

One approach to designing ITS systems is to leverage software patterns, a software technique for identifying and describing reusable, successful solutions to software problems. In reviewing sixty-six papers on ITS systems, Devedzić [33] discovers seven ITS patterns. Of interest to our work was the *application agent* pattern, which describes a layered approach to embedding software agents into existing applications. Several papers discussed here, in addition to this thesis, implement such a pattern. Devedzić [33] stresses that designers of different ITS architectures use these patterns in most cases without being aware of their existence, and such *implicit* pattern use is prominent in ITS architectural design.

Other researchers have investigated viewing ITS design from a software engineering perspective. Keeling [65] proposes a methodology that separates domain-independent and domain-dependent modules of the tutoring system, thereby reducing the involvement of knowledge engineering. The need for systematic ITS development strategies is also well known [118]. Hinostroza *et al.* [45], after observing several ITS projects, have concluded that attempts to dictate student usage models often fail and that software is used in unintended ways. Patel and Kinshuk [96] concur, and thus one can gather that good ITS design involves creating “cognitive tools” that allow individual students to learn each in their own way ([45], [75], [93] and [96]).

Paiva *et al.* [93] notes that most ITS architectures “fail to capitalize on one of the major advantages of such multi-agent systems, “the independence and reusability of agents.” Most ITS designs result in domain dependence [40]. Paiva attempts reusability through the creation of a framework for agents to communicate. This Pedagogical Agents Communication Framework, or PACF, allows for the creation of a heterogeneous

community of interacting agents [93]. Paiva considers three kinds of agents: *tutor agents*, which are responsible for the learning material, *pedagogical interface agents*, responsible for interfacing with the learner, and *domain expert agents*, responsible for the domain knowledge. Paiva's system has proven successful in the creation of learning environments that are reusable across the domains of astronomy and math. Galeev *et al.* [40] describes an as yet untested system for computing the difficulty of learning content in a domain-independent way. This parametric model incorporates student feedback in evaluating the optimal difficulty in learning content.

The lack of a well established developing environment or programming language for ITS and MAS software is a challenge to their adoption by education software developers today [130], [118]. However, there are still several attractive programming languages to facilitate MAS/ITS development. Scripting languages like TCL are commonly used in many projects [130], because they allow for rapid development and easy Internet deployment. Traditional object-oriented programming languages, such as C++ and Java, are also popular [130]. Java is especially popular because of its support for object serialization, native threading, and Internet and Web compatibility. Other projects use more agent and ITS-specific systems such as AGLess [7] and KQML [93]. Both of these languages are well suited for agent development, but lack the programmer base and interoperability of languages such as Java [130].

5.2 Data Design

The literature shows many monolithic software designs that provide student resources and tutoring in one unit. This research offers an alternate approach, separating the online resource design and interface from the teaching strategy, design, and interface.

While users of the online environment and the LS agents interact with an intelligent, unified learning environment, the actual design of PIVoT is rather complex. Several independent modules interact to provide complex behavior leveraging domain-independent design principles.

In order to have a sound architectural design, the requirements for information availability, soundness, and consistency are vital. We will discuss the information descriptors or metadata and the ontology of the system.

5.2.1 Metadata

Hodge [46] defined metadata as “structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use or manage an information resource.” By addressing the “who, what, where and why” of information, metadata allows intelligent agents (both human and software) to use this information in the searching process. Several educational metadata standards exist such as IMS [55], the Dublin Core [122] and SCORM [1]. These standards not only assure interoperability between educational systems, but also allow intelligent tutoring systems that work with standard metadata formats compatibility with content not even in existence at the time of their creation and deployment. PIVoT was designed to adhere to the requirements of one such metadata standard, the IMS standard, an extension of the Dublin Core.

The IMS metadata standard, like others, provides a structured framework to describe all instructional media items. While the standards are broad and support many different fields, each with different educational applications, PIVoT focuses on the primary ones that affect information retrieval the most. In particular, metadata in PIVoT

focuses on the keywords and “key terms” used to describe the multimedia and the topics to which each item belongs. In addition, basic information such as the author, title, and description of each item is also recorded.

As one can imagine, recording metadata manually is time-consuming. While there are automatic systems for making an initial estimate at the keywords and topics for some content types, most content items require significant manual labor to be properly and fully annotated. In order to expedite and simplify the process, a logging application was previously developed in Java. This application, the PIVoT Logger, allows content experts and their assistants to quickly enter in metadata for all PIVoT-supported media types. This application simplifies the construction and maintenance of a keyword list, a topic hierarchy, and tasks specific to each media type.

5.2.2 Ontology

All metadata gathered from logging content is stored in a relational database. In ITS design, the educational content information stored in the database is referred to as the knowledge base (KB) of the system. This educational KB is different from the expert system’s LS assessment KB, which relates questions to learning styles and stores probabilities of the student’s LS model.

The educational KB is said to ontologically represent the domain. Ontology is defined as an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them [94].

While the literature discusses many projects that design their knowledge bases specifically around the structure of a particular domain, PIVoT takes a different approach [89]. PIVoT's adherence to a domain-independent metadata standard separates it from similar research. Its ontology provides the key separation between the content and the domain it represents. The metadata used in PIVoT highlights surface relationships: that is, the descriptions and relations of media to topics and other media. The content infrastructure and identification, described further below, will give a clear picture of the media structure.

5.2.3 Content Structure and Identification

The content structure, the media (section 5.2.4) and the architecture design (section 5.3) were developed prior to this research by a team at the Center for Educational Computing Initiatives [89] at MIT. This description is useful to understand the contributions of this thesis and how it plugs into the existing system.

The knowledge domain represented in a PIVoT deployment is stored in a database and accessed by a server capable of executing specialized code that arranges and presents this information. In order for attributes and media to be easily referred to in the server's Java Virtual Machine (JVM) and the database, a simple system for retrieving and referring to media and attributes was created. Both media and attributes are unified under the common term, *content*. All subclasses of media and attributes (i.e., FAQs, questions, keywords, topics) have their own identifier type. Each subclass maintains indexes of all content of its type, assigning each new media item or attribute a unique integer, known as the *unique identifier* (UID) [89].

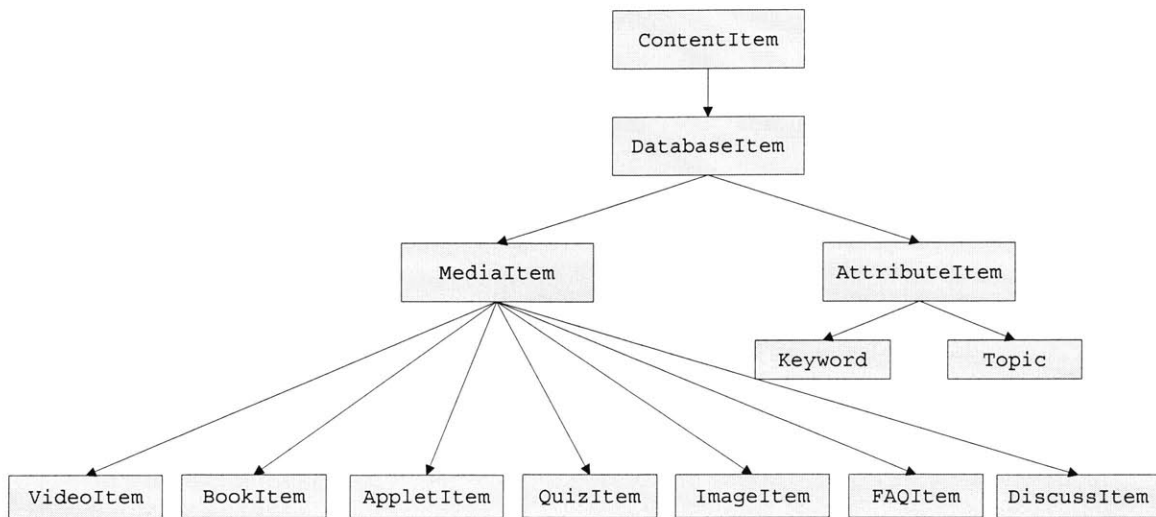


Figure 5-1: Data Interface Hierarchy

The ContentItem is used to refer to each unique media item or attribute, since it is the parent (see Figure 5-1). Each ContentItem consists of a type and UID. For example, the content identifier F102 represents an FAQ on projectile motion; a diagram (image) of force interactions between two body masses is represented by the keyword I205.

5.2.4 Media

The system represents each educational resource through the *MediaItem* class. As the name suggest, it represents a single item of educational content. The set of all MediaItems in a particular subject form the *media space*. Each MediaItem is unique and mutually exclusive with all other MediaItems. That is, a textbook represented in the

PIVoT ontology would be broken up by section into individual MediaItems, without overlap.

In order for PIVoT to function as a multimedia educational resource, it stores content of several different types. Each media type has its own subclass of MediaItem, and thus has its own set of properties useful in describing it.

5.2.4.1 Media and Metadata

All media in PIVoT, regardless of type, have certain common properties. These are based upon the IMS metadata standard described in section 5.2.1. A core subset of the IMS metadata types is used for the description of educational media.

- *Major keywords*: These terms most directly describe the concepts addressed by the content. For example, a textbook section about “kinematics of a rigid body” would have *angular acceleration*, *angular velocity* and *moment of inertia* as major keywords.
- *Minor keywords*: These terms describe the context of the item more broadly. For example, the same textbook section given above would peripherally involve the concepts of *angular momentum*, *cross product*, and *kinetic energy*.
- *Topic*: The topic represents the primary place the problem in a hierarchical course syllabus.

Keywords and topics are represented by the Java classes *Keyword* and *Topic* (see Figure 5-1), respectively. They correspond to the properties from the IMS standard.

The *Keyword* class is used to describe a single or multi-word term that is key to a specific domain, such as speed velocity for the physics domain. It is worth noting that

keywords are not mutually exclusive (i.e., it is possible for two or more keywords to refer to similar or identical topics).

The *Topic* class hierarchically arranges the broad concepts of a domain into syllabus-like structure. The universe of all topics in a domain is referred to as the *topic tree*. All content has a topic that fits in this hierarchy, with the entire subject located at the root.

It is possible for topics and keywords to have the same name, such as the keyword *acceleration* and the topic *Acceleration*. The differentiation is that the keyword is used wherever content refers to a particular concept, while a topic is used whenever a particular point in the curriculum is reached. Many media items may have “acceleration” as a key concept, but very few of these items appear in the course at the point in which “Acceleration” is introduced and discussed.

5.2.4.2 Property Classification

Some properties are specific to certain media types. For example, an electronic version of a textbook section has the beginning and end page numbers as additional descriptors. For video segments, the duration of the clip would be a useful property. However, certain key properties, such as title and description, apply to all media types. To implement this, these properties are specified in the *MediaItem* class, parent to all individual media types. Since it is impossible to create a generic “MediaItem,” the class is declared abstract – that is, only subclasses of *MediaItem* may be instantiated. For most purposes, the work described in this thesis primarily focuses on these common properties, since they allow for the greatest reuse across subjects and educational content types.

5.3 System Architecture

This section discusses the three-zone network abstraction model used for PIVoT. The system is a client-server application distributed across several machines. The model has two abstraction barriers between three zones: *client*, *server* and *backend* (see Figure 5-2). The client, using a standard Web browser and ubiquitous video application, makes requests to three kinds of servers: the web server, the video server, and other external servers. The web server runs a JVM to support all dynamic page generation; this machine is often referred to hereafter as the WS/JVM. All users of the PIVoT system are required to log in and identify themselves. These connections are secured by using certificates and the HTTPS protocol. The web server's JVM processes requests from the client and in turn queries the backend database via JDBC (Java Database Connectivity) and SQL (Structured Query Language).

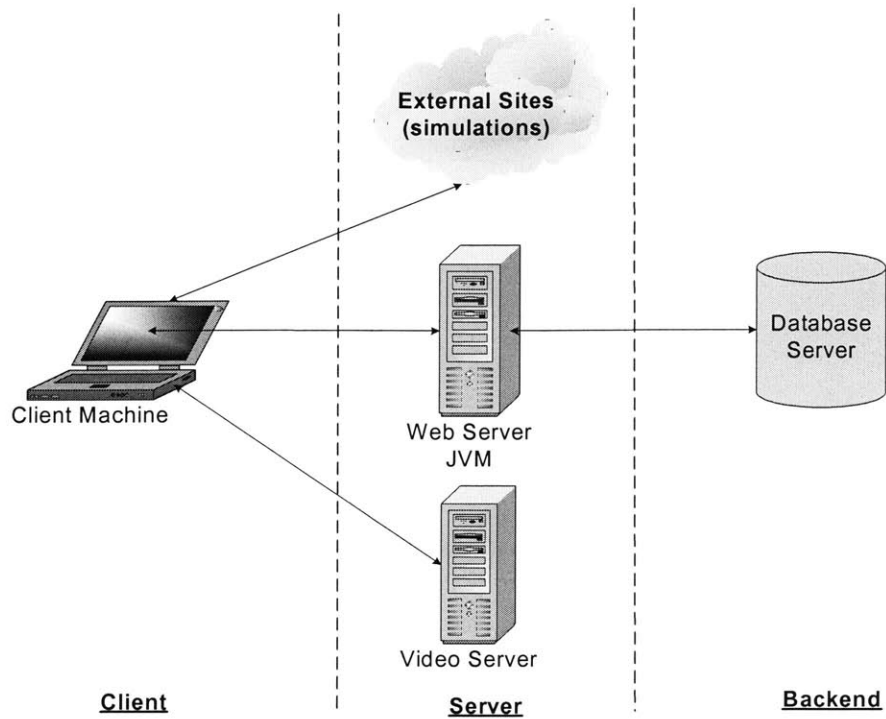


Figure 5-2: PIVoT Application Network Model

For many content types, the client is redirected to other servers, including some external to the PIVoT-maintained servers. Some simulations, for example, are located on servers elsewhere on the Internet. The database contains the URLs of these educational resources, and result pages list these links. To view these, the client clicks on these links and is then connected to an external site. The client is similarly referred to video content, located on a separate PIVoT-maintained server that communicates using a streaming-media protocol. The client receives responses from the server containing URLs that launch a separate video client (RealPlayer, in this case). This ubiquitous, multi-platform, freely available client software is used for all video streaming content.

While this discussion refers to the three PIVoT-maintained applications (database, web server and video server) as residing on separate machines, this is not a necessary fact. In PIVoT's current deployment, the web server and database reside on the same machine. This improves performance with minimal impact on modularity, since the database, JVM and server applications are isolated and "sand-boxed" by the UNIX operating system on which they all run. The video server (VS) still resides on a separate machine, on a separate MIT subnet, for network performance reason; the bandwidth and memory needs of the video server are vastly different from the needs of the WS/JVM and database.

5.3.1 Technology and Programming

The developers of PIVoT desired an environment that is both easy to develop in and platform independent. While several server specific solutions were developed, the Java-based Servlet API provides a platform independent efficient solution that has been adopted by many Web developers.

The Servlet API offers a platform-independent computing environment that mimics CGI (Common Gateway Interface) with support for persistence across client requests through cookies. Briefly, CGI is a solution for web server calls that generates dynamic content according to the request. It is inefficient, however, since each request requires a new process to be created. Cookies are a collection of information, stored on the local computer of a person using the web, used mainly by servers to identify users who have previously registered or visited the site. Because the Servlet API provides a full Java environment, programmers can leverage easy database connectivity, object-oriented persistence, and an existing code base.

The data is stored in an Oracle database version 9.2. Servlets and the Oracle database interact very easily. The Oracle Servlet Engine (OSE) is a built-in web server with an integrated Servlet engine running inside Oracle. OSE executes Java Server Pages, Servlets and Java Stored Procedures.

5.3.2 Server-Database Interaction

The logged information about a domain is stored in a relational database. Queries to the database are made through SQL. Certain web pages require several queries to the ontology, which naively could result in many database queries per web-server request. For that reason, one must take special care to reduce round-trips to and from the database by the web server.

Besides reducing the number of database queries, the database's accessibility should be transparent in the virtual machine. PIVoT uses an object-oriented class hierarchy to abstract away the details of database access. This approach has several advantages. First, this abstraction removes the intricate details of SQL programming from the higher-level logic of PIVoT and the multi-agent system. Second, changes to the implementation details, including the brand of database, can be done without changing the application-level code. Finally, similarities between the different content types result in several similar queries being executed. Using an object-oriented layer to abstract database details increase code-reuse, which reduces the possibility for bugs.

Abstracting away the details of database queries, however, impacts the ability to reduce database-server load. Each content item has many properties that can be retrieved by "get" methods in the Java API. When a row corresponding to an item is accessed via an SQL query, several columns from the table can be efficiently retrieved together, but

not all properties are always needed. On the other hand, if each “get” method results in a trip to the database, performance may be reduced.

In order to solve the database-JVM access problem, earlier versions of PIVoT developed used a tiered caching approach. It consisted of pre-loading the most basic properties of all media and attributes on initialization of the server, namely “name” and “UID.” Other fields are retrieved into a volatile cache memory that reduced the number of round trips to the database.

5.3.3 Serialization

All content stored in the database is retrieved as an instance of the DatabaseItem class. This abstract class unifies both media and attribute branches, and extends from the token-like ContentItem class (see Figure 5-1). The DatabaseItem class prevents other classes from constructing instances of database items. Instead, the class uses public retrieval methods to access instances already constructed by the API. This means that each instance of DatabaseItem is the only instance in virtual memory with that content type and UID. This uniqueness prevents wasteful construction of duplicate objects, as well as duplicate requests to the database for information already stored in memory. This uniqueness of database objects in memory does, however, pose a problem for object serialization.

The application written using the multi-agent system has the ability to save and remember state from session to session. This is accomplished with Java’s serialization API. Java objects, however, do not automatically support serialization. Whenever an object is de-serialized, it becomes a new object, constructed from the serialized form. As

such, users can create copies of an existing instance in memory by serializing it and then de-serializing the stored object. In this case, both the original and the previously stored version would exist in memory at the same time. Certain objects in PIVoT control the construction of objects and prevent copies from being made. For example, database items require that only one instance with a given content identifier may exist in memory at the same time. For this reason, traditional serialization, which copies the fields of an object to storage for later restoration, cannot be allowed to occur.

To overcome this restriction and allow content to be serialized, PIVoT was designed to extend the object stream classes for serialization and de-serialization, allowing them to substitute certain instances of objects with a token upon serialization. This token uniquely identifies the serialized object, but does not contain the object it refers to itself. Upon de-serialization, when a token is retrieved from the stream, it is automatically substituted with the unique instance already in memory, preventing duplicate instances from being created.

5.4 Multi-Agent System Architecture

The multiple agents (see section 4.2) tasks are first, to observe the student's online actions and use them for updating the student's model, and second, to recommend educational media matching the student's learning style when he or she browses or searches for new information. In order to make communication channels efficient, the architecture was designed for essential interaction between agents, the student, the server and the database. Figure 5-3 shows a UML use case of the typical interaction between student and multi-agent system.

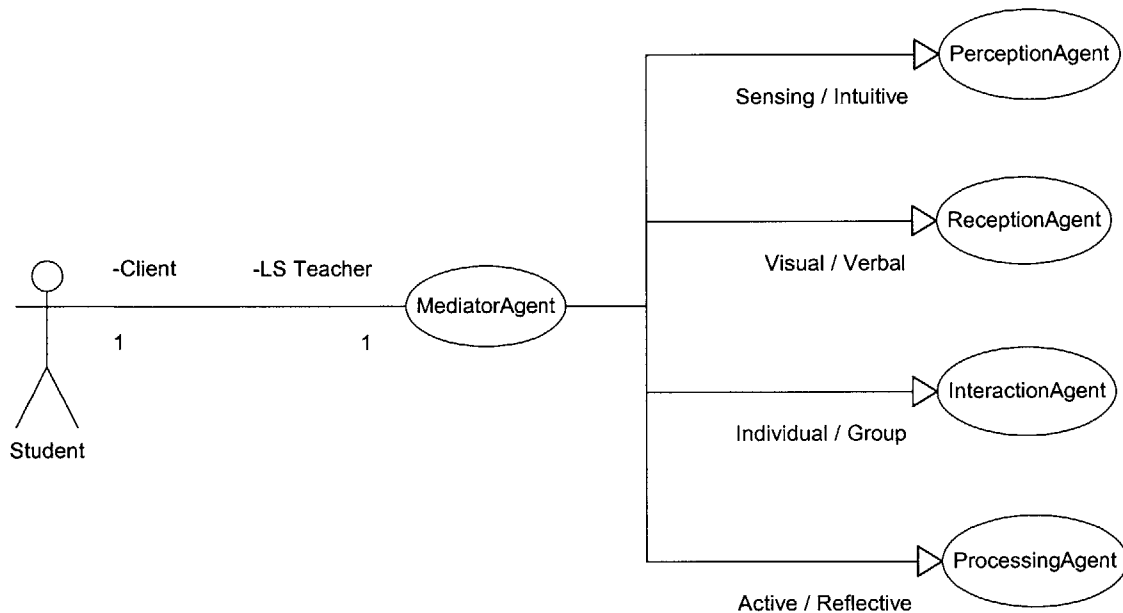


Figure 5-3: Student - MAS Interaction

5.4.1 Expert System Integration

The initial questionnaire administered to the student is constructed as a separate entity of the MAS, since it is used only at first encounter. The results of the questionnaire are stored in the same database that the MAS will be accessing in order to model the student's learning style. These values act as an initial estimate of the student's model. Notably, the values that will be stored are the student's ID, the date and time when the questionnaire was administered, and the probabilistic values of the various learning style dimensions.

Although most of the expert system classes used for building the questionnaire are not used by the MAS, the inference mechanism and the Bayesian posterior probability calculations are built as separate classes; these will be used by the multiple expert agents

for model calculations, specifically when they notice a student activity (mouse click or search query) of relevance to a change of the learning style preference. With that said, the subsequent sections focus on the MAS's interactions with the user.

5.4.2 Agents' Observations of Student Activity

The student's online behavior is used to assess his or her preferences. The agents are interested in mouse clicks and the time spent on a certain medium. For example, a student's mouse click on a video segment will fire an event that agent Visual-Verbal (VV) will interpret as an interest in visual activities. This activity is used to update the probabilistic belief of agent VV about the student's profile in the visual-verbal learning style dimension. These online activities will enable the agents to follow and adapt to the student's changing learning style.

5.4.3 Online Student Activity

The main purpose of the PIVoT interface is to find and retrieve instructional content. This task is decomposed into three distinct phases: *search*, *result*, and *request*. The first phase is typically the search phase, where the user browses through the PIVoT ontology by keyword, topic, or natural language query. Once a search has been entered, or a browsing action has come to a desired keyword or topic, the result phase is entered. Here, the content for a particular query is displayed according to topic and keyword relevance first, and then rearranged according to the student's learning style. If, after reviewing the result summaries for each media item, the user is unsatisfied with the results, he or she may return to the search phase. If relevant and interesting content is

found, the user clicks on the link and enters the third phase, where the content is presented. In the third phase video and audio are played, images are displayed, text is rendered, questions asked or answered, and discussion boards are launched. After viewing the content, the user has the option to either return to the result phase to select more instructional content, or return to the first phase to begin a new search.

In software terms, the student's actions fire events, and agents that are interested in these events register to receive a notification of action. Therefore, in order to save communication bandwidth, the event is fired once, sent to a mediator agent residing on the server, which in turn will redistribute the events to interested agents (see Figure 5-3). Agents' actions are then independently processed, and each agent will update the student's profile. The mediator agent will log the student's activity for future reference and in order to keep a history of the student's changing interests.

5.4.4 Student Profile and Database

The student's profile is stored in the backend database in order to keep the student's profile persistent across sessions. If the student logs off and back on, the agents will only need the student's user name to be able to retrieve the student's most up-to-date profile. They can then make educational media recommendations when required to.

The mediator agent also appends all the student actions to existing tables in the database. The information stored when events are fired consists of the date and time, the user ID, and the educational content's UID. The latter will be used to retrieve the medium format and observe the student's progress. By observing the time difference between two events, the duration can be calculated. This difference is sometimes problematic, since

the system cannot be sure whether the student is using the material during that time, or instead the user is preoccupied with some other online or offline activity.

5.5 Summary

The Multi-Agent System design has the advantage of being plug-able into multiple educational systems architectures. As long as the metadata provides an easy way of uniquely identifying the media format, and as long as the student's actions can be observed, the MAS can update a student LS profile and make recommendations of educational media to present to the student. This makes the system domain-independent and provides a demand-based recommendation, which minimizes intrusion into the student's learning experience.

Chapter 6 – Conclusions and Recommendations

This last chapter reviews the major technological, scientific and pedagogical contributions of this thesis and summarizes the findings. It then follows with a discussion of areas of future work that this research opens up. The chapter concludes with an analysis of how this research advances the domains of intelligent online learning and educational technology.

6.1 Contributions

The primary contribution of this research is an online personalization system that delivers educational media matching the students' individual learning styles. This system is a combination of several technologies, building upon the literature on multi-agent systems, artificial intelligence, expert systems, conflict resolution, voting theory, cognitive science, education, and learning theory.

6.1.1 Technical Contributions

The learning style assessment system provides a probabilistic model of the student's learning style. This expert system is a tool that is independent of the educational resource system and can act as a standalone application. In addition, the expert assessor system, which consists of an adaptive questionnaire, can be easily modified to account for new LS dimensions as long as an expert can estimate the relevance of questions to a learning style type (see section 3.6).

This first part can be used as a tool for the stochastic modeling of a student's learning style. An instructor or an adaptive teaching system can then use this model to tailor the instructional material accordingly. The questionnaire was able to categorize students' learning styles by using an average of approximately seventeen questions. The sequential Bayesian approach is more efficient and mathematically more rigorous than the commercially available learning style assessment questionnaires.

The expert system stores an initial student model, which is later used by a multi-agent system for online personalization of educational media. The agents are intelligent software programs, with each having knowledge about a specific learning style dimension. They have the power to observe the students' online behavior in order to constantly update the student model. The MAS considers the students' actions as expressions of their varying interests. For example, the action of "clicking on a video segment" is used to update the student's model and his/her interest in visual activities.

The multi-agent approach allows the consideration of various learning style dimensions since each agent is an expert in a specific dimension. This approach accounts for a multi-dimensional learning preference and adapts to time-varying learning styles.

One of the MAS's advantages is its expandability; new agents can be easily created after loading them with the knowledge about their specialized learning style dimension, while existing agents can be removed if they are not relevant to a particular learning environment.

6.1.1.1 Plug-in system

An essential contribution of this approach is the ability of the system to plug into existing web-based education systems. However, descriptive metadata of the learning material is required for the system to work. This metadata allows the agents to extract the content and format information they need in order to recommend the educational media that match the student's style. In order for the multi-agent system to add value, the educational website needs to have a diversity of resources relevant to topics. In the case of scarce resources related to a single topic, the student's learning style profile would not significantly contribute to choosing or ranking the resources matching the student's preferences.

6.1.1.2 Conflict Resolution

This research contributes to the literature on the resolution of conflicts among advising agents. The agents provide the mediator agent with their ranked list of preferred educational media. The mediator then inputs these recommendations into a ballot and provides a resulting cycle-free ranked list that incorporates all the agents' input. This result can be used in a variety of activities; this research's aim was an application to the returned search results. When the student searches for keywords, the results are returned

according to topic and keyword match (see section 5.2.4). The mediator agent's ranked list is used to reorder the search results to account for LS preferences (see section 4.8). By quantitatively assessing learner preferences, students can receive automatically personalized assistance with minimal delay, encouraging the use of valuable web-based resources.

6.1.2 Pedagogical Contributions

After careful analysis and comparison of the available learning style models, we found that these were developed for classroom environments. The problem with that finding is the transferability to online environments. In such new environments, different learning methods and materials are available. In addition, some restrictions and limitations surface and therefore hinder the application of these models. This thesis addressed and solved this issue by selecting from the various models the learning style dimensions that are applicable to the available online resource. In particular, the approach chose the dimensions that could be used to describe the available online material and to compare the media along that dimension. For example, a video segment is more attractive to visual students rather than verbal ones, involves a more visual activity than a problem set does.

This approach yielded four learning style dimensions (see sections 3.3 and 3.4) appropriate for the PIVoT environment that includes seven different educational formats and activities. Had the website had different resources, the dimensions would have to be changed to adapt to the new environment. In summary, there is no one learning style

model generally applicable to online environments; the proposed solution is a customized model to the available resources.

The learning styles addition to an online environment stemmed from an actual need. The application to PIVoT was an excellent test bed since Niemczyk [89] had noticed in his thesis that students, "...prefer to go to the advanced search page and customize the media type..." Therefore, this learning style incorporation aimed to help the students in delivering the media that suited their learning preferences.

6.2 Future Research

This research opens up new opportunities for future research in the areas of educational assessment, expert system enhancement, teaching strategies, and intelligent educational environments.

6.2.1 Educational Assessment

The system, as it stands, delivers online material to the students according to their stochastically-assessed learning style model. However, future research needs to evaluate the effectiveness of this system in terms of grade improvement, online time spent, relevance of search results to students' expectations, and qualitative assessment of the performance of students taking the class. The qualitative assessment can be done in the format of focus groups, where students provide feedback on the system usability and usefulness.

6.2.1.1 Grade Improvement

By personalizing an online resource, the student's grades can potentially be affected. Future assessment studies can measure the grades difference between students using the learning styles personalization system and those using the website without the customized search results. If this change is statistically significant, one can go further to analyze the students' grades for various grade intervals and to identify the groups that benefit the most.

6.2.1.2 Online Time

The students' online time has a high likelihood of change. Initial assessment of students' online usage time shows both duration increase and decrease with the addition of the multi-agent system. We give some reasons that show the need for a study of online time; a duration increase can be attributed to an affinity for the system, or to an excessive time spent going through unnecessary extra material, which can be repetitive or overlapping and therefore inefficient. A duration decrease can be credited to a student's alienation to the system or to greater efficiency in finding the information due to the presence of the multi-agent system.

6.2.1.3 Relevance of Search Results

Since the multi-agent system delivers customized search matches, future research needs to study the frequency of users opting for variously ranking search results. One can study if the student is clicking on the highest-ranking search results, or rather opting for the lower-ranking ones. The dynamic ranking multiplier (see section 4.8) was used to

adjust rankings in the latter case by decreasing the effect of the learning style profile on search results. However, one can make that multiplier a static one and observe whether learning style incorporation does in fact help the student find learning materials that match his or her learning style.

6.2.2 Expert System Improvement

The expert system currently includes data that relates answers to mundane questions to learning style preferences. This data comes in the form of probabilistic values of the student's answer given his or her belonging to a learning style type. These probability values were input based on literature search and our own judgment; therefore, they carry a degree of uncertainty. After several runs of the system on various student populations, one can use the student's answers and their LS profile to go back to these probability values and validate the accuracy of these likelihood values. In addition, one can also ask the students to assess their own LS profiles and to estimate their learning preferences. This self-assessment can be used to validate the expert system's results and subsequently to verify the questions' likelihood values.

In addition, the initial values of a student profile are set equally for each learning style type. In other words, we start with no information about the student. In order to further decrease the questionnaire time and number of questions asked, one can use the collective profile of students as prior knowledge about the students' LS. However, this holds in the case where students generally belong to a homogeneous population from year to year. If the students have a common trend of learning styles, this collective data

would prove to be very effective in estimating the students' LSs prior to joining the course.

6.2.3 Teaching Strategies

The personalization system presented in this research solved the case of search matches, but left the task of choosing the sequence of learning materials to the student. The literature is replete with teaching strategies, with Kolb's learning strategy [69] and the 4MAT method [82] as prime examples of applied strategies and curriculum development. The major issue in applying these strategies to online material is the design phase. The learning material design should follow a certain strategy from its inception steps. The simple fact of having online material is insufficient to administer a course with a sound strategy.

The problem that this research solved is the application of a LS personalization system to the case in which the material exists in an abundant amount, but without clear relations among learning materials.

6.2.4 Intelligent Educational Environments

This thesis presented a way to make educational environments more intelligent and to customize the learning materials to the students' learning styles. Future contributions to this area could come in the form of assistance and encouragement to the students. Even though the system is not used for course grading, future work can use

positive and negative reinforcement strategies to encourage them to use the resource to their advantage.

6.2.5 Corporate Applications

In large companies, it is often necessary to train a large segment of the firm's employees through videotape-based lectures and paper-based testing. These training sessions are costly and time-consuming. By providing intelligent online learning resources, the training time can be cut down by providing learning materials that match the employees' learning styles. Since this system can be plugged into an existing online resource, future work can test the effectiveness and efficacy of intelligent teaching in the corporate realm.

This research's findings can be applied to the maturing *online search* business. The current *search engines* provide thousands and millions of search matches when submitting various queries. However, there is no need to display that many matches since no person would have the time or the patience to go through all of them. The main reasons behind the number of matches are the abundance of online material, the number of domains a word belongs to (semantic relevance), and the ignorance of the search engine about the preferences and the areas of interest of the user. This research solved a problem in the field of online education; however, it can be applied to search engines, by modeling the users' preferences, making the "search" experience more personalized, and by contributing to a decrease of search matches and an increase of search relevance.

6.3 Conclusions

This research brought many contributions to the art and science of intelligent online tutoring systems. The expert system's Bayesian approach for assessment of learning styles was an innovative way of approaching the problem of learning style assessment. The inclusion of a decision-making process based on value of information proved to be a highly efficient tool for LS assessment questionnaires, compared to the simpler approaches of the commercially available ones.

The multi-agent system used for recommending learning materials based on learning styles was also an innovative application, combining artificial intelligence, cognitive styles, and learning theory. Within the multi-agent system, the voting algorithms applied to resolve the conflicts among the agents were an original application of voting theory to the realm of multi-agent systems.

As stated above, this research opened up venues for future research in the areas of artificial intelligence and education. The future endeavors of assessment and advancement of the system may help further our understanding of the learning process and the role that science and technology can play to take us a step closer to understanding the rich and complex ways in which people learn.

This page is intentionally left blank

Appendix A

Conditional Likelihood of Educational Media Over LS Dimensions

The distribution of conditional likelihood of the educational media over the various learning style dimensions is separated into matrices for each dimension. These values will be used for the agents' ranked recommendations given the student's probabilistic profile in each dimension. Each agent will be able to rank the educational media from 1 to 7, and this list will be entered into the ballot to compete with the other agents' recommendations.

The conditional likelihood values were estimated by knowing the activity involved with each medium, and the appeal to learners of each type.

Distribution of Conditional Likelihood of Media Over Visual-Verbal LS

$p[\text{medium} \text{type}]$	Video	Simulation	Problem	Text	Discuss	FAQ	Diagram
Visual	0.25	0.3	0.1	0.06	0.03	0.06	0.2
Verbal	0.06	0.04	0.15	0.2	0.25	0.2	0.1
Neutral	0.08	0.04	0.25	0.2	0.03	0.2	0.2

Distribution of Conditional Likelihood of Media Over Active-Reflective LS

$p[\text{medium} \text{type}]$	Video	Simulation	Problem	Text	Discuss	FAQ	Diagram
Active	0.1	0.15	0.3	0.05	0.25	0.08	0.07
Reflective	0.15	0.05	0.1	0.3	0.05	0.15	0.2
Neutral	0.3	0.15	0.8	0.8	0.12	0.15	0.12

Distribution of Conditional Likelihood of Media Over Sensing-Intuitive LS

$p[\text{medium} \text{type}]$	Video	Simulation	Problem	Text	Discuss	FAQ	Diagram
Sensing	0.1	0.07	0.3	0.25	0.1	0.1	0.08
Intuitive	0.15	0.4	0.05	0.05	0.05	0.05	0.25
Neutral	0.2	0.08	0.07	0.15	0.2	0.15	0.15

Distribution of Conditional Likelihood of Media Over Individual-Group LS

$p[\text{medium} \text{type}]$	Video	Simulation	Problem	Text	Discuss	FAQ	Diagram
Individual	0.08	0.1	0.2	0.3	0.01	0.06	0.25
Group	0.15	0.1	0.05	0.05	0.4	0.2	0.05
Neutral	0.2	0.3	0.2	0.05	0.05	0.1	0.1

References

- [1] Advanced Distributed Learning, "SCORM: Shareable Content Object Reference Model," version 1.2.
- [2] Albacete, P. and VanLehn, K., "The Conceptual Helper: An Intelligent Tutoring System for Teaching Fundamental Physics Concepts," in *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference*, ITS, Montreal, Canada, 2000.
- [3] Anderson, J., *The Architecture of Cognition*, 1983, Reprint, Mahwah, NJ: Lawrence Erlbaum Associates, 1996.
- [4] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T., "WebWatcher: A Learning Apprentice for the World Wide Web," *Proceedings of AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments*, Stanford, CA, 1995.
- [5] Arrow, K., *Social Choice of Individual Values*, Yale University Press, 1963.
- [6] Arrow, K. and Raynaud, H., *Social Choice and Multicriterion Decision-making*, Cambridge, Mass., MIT Press, 1986.
- [7] Badjonski, M., Ivanović, M. and Budimac, Z., "Intelligent Tutoring System as Multiagent System," *IEEE Conference on Intelligent Processing Systems*, vol. 1, pp. 871-875, 1997.
- [8] Baylor, A., "Cognitive Requirements for Agent-Based Learning Environments," *IEEE International Conference on Advanced Learning Technologies*, pp. 462-463, 2001.
- [9] Beaumont, I., "User Modeling in the Interactive Anatomy Tutoring System ANATOMTUTOR," *User Modeling and User-Adapted Interaction*, vol. 4 (1), pp. 21-45, 1994.
- [10] Beck, J. and Woolf, B., "Using a Learning Agent with a Student Model", *Intelligent Tutoring Systems*, pp. 6-15, 1998.

- [11] Belkada, S., Cristea, A. and Okamoto, T., "Measuring Knowledge Transfer Skills by Using Constrained-Student Modeler Autonomous Agent," *IEEE International Conference on Advanced Learning Technologies*, pp. 375-373, 2001.
- [12] Bellman, R., *An Introduction to Artificial Intelligence: Can Computers Think?*, Boyd and Fraser Publishing Company, 1978.
- [13] Bernardo, J., "An Assignment Approach to Choosing R&D Experiments," *Decision Sciences*, vol. 8, pp. 489-501, 1977.
- [14] Binemann-Zdanowicz, A., Schewe, K. and Thalheim, B., "Adaptation to Learning Styles," *IEEE International Conference on Advanced Learning Technologies*, pp. 121-125, 2004.
- [15] Bollen, J., "Cognitive Complexity vs. Connectivity, Efficiency Analysis of Hypertext Networks, in "The Evolution of Complexity," Heylighen, Fl, Bollen, J., and Riegler, A. (eds), Kluwer Academic Publisher, pp. 345-368, 1999.
- [16] Borda, J-C, *Mémoire sur les Elections au Scrutin*, Histoire de l'Académie Royale des Sciences, Paris, 1781.
- [17] Borg, M. and Shapiro, S., "Personality Type and Student Performance in Principles of Economics", *Journal of Economic Education*, vol. 27, pp. 3-25, 1996.
- [18] Boulay, B., "Can We Learn from ITSs?" in *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference*, ITS, Montreal, Canada, 2000.
- [19] Boyle, C. and Encarnacion, A., "MetaDoc: An Adaptive Hypertext Reading System," *User Modeling and User-Adapted Interaction*, vol. 4(1), pp.1-19, 1994.
- [20] Brenner, W., "Agents as Tools of the Information Society," in *Intelligent Software Agents*, by Brenner, W., Zarnekow, R., and Wittig, H., in cooperation with Schubert, C., translated by Rudd, A., pp. 7-18, Berlin: Springer, 1998.
- [21] Brookfield, S., *The Skillful Teacher*. San Francisco: Jossey-Bass, 1990.
- [22] Brusilovsky, P. et al., ELM-ART: An intelligent tutoring system on World Wide Web, In *Intelligent Tutoring Systems, Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96*, Lecture Notes in Computer Science, Vol. 1086, Eds. Frasson, C., et al., Berlin: Springer, pp. 261-269, 1996.
- [23] Brusilovsky, P. and Pesin, L., "ISIS-Tutor: An Adaptive Hypertext Learning Environment," *Proceedings of JCKBSE, Japanese-CIS Symposium on knowledge-based software engineering*, Eds. Ueno, H., Stefanuk, V., Pereslavl-Zalesski, Russia, 10-13, pp. 83-87, 1994.

- [24] Brusilovsky, P., Intelligent Tutor, "Environment and Manual for Introductory Programming," *Educational and Training Technology International*, vol. 29(1), 26-34, 1992.
- [25] Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia", *User Modeling and User-Adapted Interaction*, vol. 6, pp. 87-129, 1996.
- [26] Carver, A., Howard, R. and Lane, W., "Enhancing Student Learning Through Hypermedia Courseware and Incorporation of Student Learning Styles," *IEEE Transactions on Education*, vol. 42(1), p. 33-38, 1999.
- [27] Condorcet, Marquis de, An Essay on the Application of Probability Theory to Plurality Decision-Making: An Election Between Three Candidates, Eds. Sommerland and Mclean, pp. 69-80, 1989.
- [28] Consequence Management Program Integration Office, "Sharable Content Object Reference Model (SCORM) in the Eyes of Instructional Designers: Past and Future," Carnegie Mellon University, 2001.
- [29] Cook, W. and Seiford, L., "Priority Ranking and Consensus Formation," *Management Science*, 24, pp. 1721-1732, 1978.
- [30] De Bra, P., "Teaching hypertext and Hypermedia Through the Web," *World Conference of the Web Society Proceedings*, pp.130-135, 1996.
- [31] de La Passardière, B. and Dufresne, A., "Adaptive Navigational Tools for Educational Hypermedia," in Computer Assisted Learning, *Proceedings of ICCAL'92, 4-th International Conference on Computers and Learning*, Ed. Tomek, I., Berlin: Springer, pp. 555-567, 1992.
- [32] Dede, C., Brown, T. and Whitehouse, P., "Designing and studying learning experiences that use multiple interactive media to bridge distance and time," Draft Paper for, Current Perspectives on Applied Information Technologies, (1) Distance Education, Eds. Vrasidas, C. and Glass, G., 2002.
- [33] Devedzić, V., "Applying Patterns to ITS Architectures," in *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference*, ITS, Montreal, Canada: 2000, Lecture Notes in Computer Science; 1839, Eds. Gauthier, G., Frasson, C. and VanLehn, K., pp. 123-132, Berlin: Springer, 2000.
- [34] Dewey, J., How We Think, New York: Heath, 1933.
- [35] Dunn, R. Dunn, K., and Price, G. Learning Style Inventory Research Manual Price Systems, 1985.
- [36] Eliot, C. and Woolf, B.P., "An Adaptive Student Centered Curriculum for an Intelligent Training System," *User Modeling and User-Adapted Interaction*, vol. 5, pp. 67-86, 1995.

- [37] Felder, R., "How Students Learn: Adapting Teaching Styles to Learning Styles," *Proceedings of Frontiers in Education Conference, ASEE/IEEE*, Santa Barbara, CA, p. 489, 1988.
- [38] Felder, R. and Silverman, L., "Learning and Teaching Styles in Engineering Education", *Engineering Education*, vol. 78(7), pp. 674–681, 1988.
- [39] Fleming, N., "I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom," in Zelmer, A., (ed.) *Research and Development in Higher Education, Proceedings of the Annual Conference of HERDSA*, 18, pp. 308 – 313, 1995.
- [40] Galeev, I., Sosnovsky, S. and Chepegin, V., "ITS Design Technology for the Broad Class of Domains," *IEEE International Conference on Advanced Learning Technologies*, pp. 431-432, 2001.
- [41] Gardner, H. and Hatch, T., "Multiple Intelligences Go to School: Educational implications of the Theory of Multiple Intelligences," *Educational Researcher*, vol. 18(8), pp. 4-9, 1989.
- [42] Gasser, L., "MAS Infrastructure: Definitions, Needs, and Prospects," in *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, International Workshop on Infrastructure for Scalable Multi-Agent Systems*, Eds. Wagner, T. and Rana, O., pp. 1-11. Barcelona, Spain: 2000, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, 1887, Berlin: Springer, 2001.
- [43] Gonschorek, M. and Herzog, C., "Using Hypertext for an Adaptive Help System in an Intelligent Tutoring System," *Proceedings of AI-ED, 7th World Conference on Artificial Intelligence in Education*, Ed. Greer, J., Washington, DC, pp. 274-281, 1995.
- [44] Heift, T., and Nicholson, D., "Theoretical and Practical Considerations for Web-Based Intelligent Language Tutoring Systems," in *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference*, ITS, Montreal, Canada, 2000, Lecture Notes in Computer Science, 1839, eds. Gauthier G., Frasson, C., and VanLehn, K., pp. 354-362, Berlin: Springer, 2000.
- [45] Hinostroza, E., Rehbein, L., Mellar, H. and Preston, C., "Developing Education Software: a Professional Tool Perspective," *Education and Information Technologies*, vol. 5(2), pp. 103-117, 2000.
- [46] Ho, T., Hull, J. and Srihari, S., "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, 1994.
- [47] Hodge, G., National Informational Standards Organization.

- [48] Hohl, H. et al., "HyperAdapter: An Adaptive Hypertext System for Exploratory Learning and Programming," *User Modeling and User-Adapted Interaction*, vol. 6 (2-3), pp. 131-156, 1996.
- [49] Hong, H. and Kinshuk, "Adaptation to Student Learning Styles in Web Based Educational Systems," in L. Cantoni & C. McLoughlin (Eds.), *Proceedings of ED-MEDIA - World Conference on Educational Multimedia, Hypermedia & Telecommunication*, pp. 491-496, 2004.
- [50] Höök, K. et al., "A glass box approach to adaptive hypermedia," *User Modeling and User-Adapted Interaction*, vol. 6 (2-3), pp. 157-184, 1996.
- [51] <http://www.cbsnews.com/stories/2003/08/26/earlyshow/contributors/reginalewis/main570268.shtml>
- [52] <http://curricula2.mit.edu/pivot>
- [53] http://www.eduventures.com/about/press_room/03_09_04.cfm
- [54] Huitt, W., "Using the 4MAT System to Design Web-based Instruction", *8th Annual Conference: Applied Psychology in Education, Mental Health, and Business*, Valdosta, GA, 2000.
- [55] IMS Global Learning Consortium, "IMS Learning Resource Meta-Data Information Model: Version 1.2.1 Final Specification."
- [56] Jennings, N. and Wooldridge, M., *Agent Technology: Foundations, Applications, and Markets*, Berlin: Springer, 1998.
- [57] Jennings, N.R. and Woolridge, M., "Applying Agent Technology," *International Journal of Applied Artificial Intelligence*, vol. 9(4), pp. 351-369, 1995.
- [58] Jewler, A.J., Gardner, J.N., *Your College eExperience* (concise ed.), Belmont, CA, Wadsworth, p. 168, 1993.
- [59] Johnson, E. and Ganeshan, R., "Pedagogical Agents on the Web," *Proceedings of the AI-ED Workshop on Pedagogical Agents*, Kobe, Japan, pp. 48-55, 1997.
- [60] Johnson, W. and Shaw, E., "Using Agents to Overcome Deficiencies in Web-Based Courseware", *IJCAI Animated Interface Agents Workshop*, 1997.
- [61] Johnson, W., Rickel, J., Stiles, R. and Munro A., "Integrating Pedagogical Agents into Virtual Environments", *Presence*, vol. 7, p. 5, 1998.
- [62] Jonassen, D. H., *Computers in the Classroom*, Englewood Cliffs, NJ: Merrill, Keefe, J. W. (1987), in "Learning Style Theory and Practice," Reston, VA: National Association of Secondary School Principals, 1996.

- [63] Kam-Wah Wu, A., "Paradigms for Intelligent Tutoring System," *IEEE region 10 Conference on Computer, Communication, Control and Power Engineering*, vol. 1, pp. 96-99, 1993.
- [64] Kaplan, C. et al., "Adaptive Hypertext Navigation Based on User Goals and Context," *User Modeling and User-Adapted Interaction*, vol. 3, pp. 193-220, 1993.
- [65] Keeling, H., "A Methodology for Building Intelligent Educational Agents," in *Artificial Intelligence in Education: Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration, Frontiers in Artificial Intelligence and Applications*, vol. 50, Eds. Lajoie, S. and Vivet, M., pp. 46-53, Amsterdam: IOS Press, 1999.
- [66] Keeney, R. and Raiffa, H., *Decisions with Multiple objectives: Preferences and Value Tradeoffs*, Wiley, New York, 1976.
- [67] Koedinger, K., Anderson, J., Hadley, W. and Mark, M., "Intelligent Tutoring Goes to School in the Big City," *International Journal of Artificial Intelligence in Education*, vol. 8(1), pp. 30-43, 1997.
- [68] Kohler, G., "Choix Multicritère et Analyse Algébrique de Données Ordinales", Thèse de doctorat en Mathématiques appliquées, Université scientifique et médicale-Institut National Polytechnique, 1978.
- [69] Kolb, D., *Experiential Learning: Experience as the Source of Learning and Development*, Prentice-Hall, Englewood Cliffs, NJ, 1984
- [70] Kolb, D., "Learning Style Inventory-version 3: Technical specifications," TRG Hay/McBer, Training Resources Group, 1999.
- [71] Kozma, R., "Learning with Media," *Review of Educational Research*, vol. 61(2), pp. 179-211, 1999.
- [72] Lansdowne, Z., "Ordinal Ranking Methods for Multicriterion Decision Making," *Naval Research Logistics*, vol. 43, pp. 613-627, 1996.
- [73] Lee, C. and Singh, Y., "A Case-based Agent Framework for Adaptive Learning," *IEEE International Conference on Advanced Learning Technologies*, WI, pp. 235-238, 2001.
- [74] Lelouche, R., "A collection of Pedagogical Agents for Intelligent Educational Systems," *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference*, ITS, Montreal, Canada, 2000, *Lecture Notes in Computer Science*, 1839, Eds. Gauthier, G., Frasson, C., and VanLehn, K., pp. 143-152, Berlin: Springer, 2000.

- [75] Less, J., Cumming, G. and Finch, S., "Agent Systems for Diversity in Human Learning," in *Artificial Intelligence in Education: Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration*, *Frontiers in Artificial Intelligence and Applications*, Eds. Lajoie, S. and Vivet, M., vol. 50, pp. 13-20, Amsterdam: IOS Press, 1999.
- [76] Lester, J., Converse, S., Stone, B., Kahler, S. and Barlow, T, "Animated Pedagogical Agents and Problem-solving Effectiveness: A Large-scale Empirical Evaluation", *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*, 1997.
- [77] Lewin, K., *A Dynamic Theory of Personality*. New York: McGraw-Hill, 1935.
- [78] Lieberman, H., Van Dyke, N.H. and Vivacqua A., "Let's Browse: A Collaborative Web Browsing Agent," *International Conference on Intelligent User Interfaces*, 1999.
- [79] Lind, J., *Iterative Software Engineering for Multi-agent Systems: The MASSIVE Method*, *Lecture Notes in Computer Science*, *Lecture Notes in Artificial Intelligence*, 1994, Berlin: Springer, 2001.
- [80] Maes, P., "Intelligent Software: Programs That Can Act Independently Will Ease the Burdens that Computers Put on People," *Scientific American*, vol. 273(3), pp. 84-86, 1995.
- [81] Mathé, N. and Chen, J., "User-centered Indexing for Adaptive Information Access," *User Modeling and User-Adapted Interaction*, vol. 6(2-3), pp. 225-261, 1996.
- [82] McCarthy, B. and St. Germaine, C., *The 4MAT Learning Type Measure*, About Learning, 2000.
- [83] McCaulley, M., "The MBTI and Individual Pathways in Engineering Design", *Engineering Education*, vol. 80, pp. 537-542, 1990.
- [84] Mezirow, J., "A Critical Theory of Adult Learning and Education," *Adult Education* vol. 32(1), pp. 3-24, 1981.
- [85] Millán, E., Pérez-de-la-Cruz, J. and Suárez, E., "Adaptive Bayesian Networks for Multilevel Student Modeling," *Intelligent Tutoring systems, Proceedings of the Fifth International Conference*, Montreal Canada, 2000.
- [86] Murray, W., "An Easily Implemented, Linear-time Algorithm for Bayesian Student Modeling in Multi-level Trees," *Artificial Intelligence in Education: Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration*, *Frontiers in Artificial Intelligence and Applications*, vol. 50, pp. 413-420, 1999.

- [87] Myers, I., Manual: The Myers-Briggs Type Indicator, Palo Alto, CA: Consulting Psychologists Press, 1962.
- [88] Myers, I. and McCaulley, M., Manual, A Guide to the Development and Use of the Myers-Briggs Type Indicator, Consulting Psychologists Press, Palo Alto, CA, 1986.
- [89] Niemczyk, S., An Adaptive Domain-Independent Agents-Based Tutor for Web-Based Supplemental Learning Environments, Doctoral Thesis, Massachusetts Institute of Technology, 2003.
- [90] Nilsson, N., Artificial Intelligence: A New Synthesis, San Francisco, CA: Morgan Kaufmann Publishers, 1998.
- [91] Nwana, H. and Ndumu, D., "A Brief Introduction to Software Agent Technology," in Agent Technology: Foundations, Applications, and Markets, Eds. Jennings, N., Wooldridge, J., Berlin: Springer, pp. 29-47, 1998.
- [92] Ordeshook, P., Game Theory and Political Theory: An Introduction, Cambridge University Press, Chap. 2, Arrow's Result, pp. 53-71, 1995.
- [93] Paiva, A., Machado, I. and Martins, A., "Using and Re-using Agents in Multi-agent Learning Environments," in *Artificial Intelligence in Education: Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration*, *Frontiers in Artificial Intelligence and Applications*, vol. 50, Eds. Lajoie, S. and Vivet, M., pp. 750-752, Amsterdam: IOS Press, 1999.
- [94] Papaterpos, C., Nektarios, G. and Paptheodorou, T., "An Ontology for Modeling Ill-Structured Domains in Intelligent Educational Systems," *IEEE International Conference on Advanced Learning Technologies*, pp. 41-42, 2001.
- [95] Paredes, P. and Rodriguez, P., "Considering Learning Styles in Adaptive Web-based Education", *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, pp. 481- 485, 2002.
- [96] Patel, A. and Kinshuk, "Intelligent Tutoring Tools-A Problem-Solving Framework for Learning and Assessment," *Frontiers in Education Conference*, vol. 1, pp. 140-144, 1996.
- [97] Pek, P. and Poh, K., "Framework of a Decision-Theoretic Tutoring System for Learning of Mechanics," *Journal of Science Education and Technology*, vol. 9(4), pp. 343-356, 2000.

- [98] Pérez, T., Gutiérrez, J. and Lopistéguy, P., "An Adaptive Hypermedia System," *Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education*, Ed J. Greer, J., Washington, DC, pp. 351-358, 1995.
- [99] Peters, J., Jarvis, P. et al., *Adult Education*, San Francisco, CA, Ed Rogers, A., Teaching Adults, Buckingham: Open University Press, 1996.
- [100] Piaget, J. -P., *The Origins of Intelligence in Children*, International Universities Press, New York, 1952.
- [101] Prentzas, J., Hatzilygeroudis, I. and Koutsojannis, C., "A Web-Based ITS Controlled by a Hybrid Expert System," *IEEE International Conference on Advanced Learning Technologies*, pp. 239-240, 2001.
- [102] Rabbat, R., Lerman, S., "Multi-Agent System Agreement on Ranking Educational Media", submitted to *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2004, available at: <http://web.mit.edu/ralph/www/papers/MAS.pdf>
- [103] Reichmann, S. and Grasha, A., "A Rational Approach to Developing and Assessing the Construct Validity of a Student Learning Style Scale Instrument," *Journal of Psychology*, vol. 87, pp. 213-223, 1974.
- [104] Rhodes, B. and Maes, P., "Just-in-time Information Retrieval Agents", *IBM Systems Journal* special issue on the MIT Media Laboratory, vol. 39(3-4), pp. 685-704, 2000.
- [105] Ritter, S., du Boulay, B. and Mizoguchi, R., "Communication, Cooperation, and Competition Among Multiple Tutor Agents," *Artificial Intelligence in Education*, pp. 31-38, 1997.
- [106] Roselle, M. and Grandbastien, M., "Experimenting Features from Distinct Software Components on a Single Platform," *Intelligent Tutoring Systems, Proceedings of the Fifth International Conference, ITS*, Montreal, Canada, 2000, Lecture Notes in Computer Science, 1839, Eds. Gauthier, G., Frasson, C. and VanLehn, K., Berlin: Springer, 2000.
- [107] Rundle, S. and Dunn, R., "The Guide to Individual Excellence: A Self-Directed Guide to Learning and Performance Solutions," New York, *Performance Concepts International*, 2000.
- [108] Saari, D., *Basic Geometry of Voting*, Springer-Verlag, Berlin, 1995.
- [109] Saaty, T., *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1994.
- [110] Shaw, E., Johnson, W. and Ganeshan, R., "Pedagogical Agents on the Web". *Proceedings of the AI-ED Workshop on Pedagogical Agents*, Kobe, Japan, pp. 48-55, 1997."

- [111] Shortliffe, E., MYCIN: Computer-Based Medical Consultations, New York, American Elsevier, 1976.
- [112] Sims, R., Veres, J., Watson, P. and Buckner, K., "The reliability and classification stability of the Learning Style Inventory," *Educational and Psychological Measurement*, p. 46, 1986.
- [113] Snow, R., Aptitude-Treatment Interaction as a Framework for Research on Individual Differences in Learning, in *Learning and Individual Differences: Advances in Theory and Research*, pp. 13-60, Ackerman, P., Sternberg, R., and Glaser, R., New York, Freeman, W., 1989.
- [114] Trower, T., "Microsoft Agent, A Multi-Client Programming Interface for Interactive Computer Characters," in *Intelligent Agent Technology: Systems, Methodologies, and Tools*, Eds. Liu. J. and Zhong, N., pp. 379-383, 1999.
- [115] Ueno, M., "Student Models Construction by using Information Criteria," *IEEE International conference on Advanced Learning Technologies*, pp. 331-334, 2001.
- [116] Vassileva, J., "A Task-centered Approach for User Modeling in a Hypermedia Office Documentation System," *User Modeling and User-Adapted Interaction*, vol. 6(2-3), pp. 185-224, 1996.
- [117] Vassileva, T., Trajkovic, T. and Davcev, D., "Experimental Data About Knowledge Evaluation in a Distance Learning System," *Joint 9th World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, vol. 2, pp. 791-796, 2001.
- [118] Veres, J., Sims, R. and Locklear, T., "Improving the Reliability of Kolb's Revised LSI," *Educational and Psychological Measurement*, vol. 51, pp.143-150, 1991.
- [119] Veres, J., Sims, R. and Shake, L., "The Reliability and Classification Stability of the Learning Style Inventory in Corporate Settings," *Educational and Psychological Measurement*, vol. 47(4), pp. 1127-1133, 1987.
- [120] Virvou, M. and Tsiriga, V., "An Object-Oriented Software Life Cycle of an Intelligent Tutoring System," *Journal of Computer Assisted Learning*, vol. 17, pp. 200-205, 2001.
- [121] Vogel, D. and Klassen, J., "Technology-Supported Learning: Status, Issues, and Trends," *Journal of Computer Assisted Learning*, vol. 17, pp. 104-114, 2001.
- [122] Weibel, S., Kunze, J., Lagoze, C. and Wolf, M., "Dublin Core Metadata Resource Discovery," Internet Engineering Task Force, RFC 2413.
- [123] Winston, P., Artificial Intelligence, Addison-Wesley, third edition, 1992.

- [124] Wolf, C., "iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education," *ACE2003*, Adelaide, South Australia, Australia, *Conferences in Research and Practice in Information Technology*, 2003.
- [125] Woolridge, M. and Jennings, N., "Intelligent agents: theory and practice", *Knowledge Engineering Review*, vol. 10(2), pp. 115-152, 1995.
- [126] Young, H.P., "An Axiomatisation of Borda's Rule," *Journal of Economic Theory*, vol. 9, pp. 43-52, 1974.
- [127] Young, H.P., "Condorcet's Theory of Voting," *American Political Science*, vol. 82, pp. 1231-1244, 1988.
- [128] Zarnekow, R., "Development Methods and Tools," In *Intelligent Software Agents*, Brenner, W., Zarnekow, R., and Wittig, H., in cooperation with Schubert, C., Berlin: Springer, 1998.
- [129] Zarnekow, R., *Fundamental Concepts of Intelligent Software Agents*, in *Intelligent Software Agents* by Brenner, W., Zarnekow, R., Wittig, H., in cooperation with Schubert, C., translated by Rudd, A., pp. 7-18, Berlin: Springer, 1998.
- [130] Zarnekow, R. and Wittig, H., *Base Modules of Agent Systems*, in *Intelligent Software Agents*, by Brenner, W., Zarnekow, R., Wittig, H., in cooperation with Schubert, C., translated by Rudd, A., Berlin: Springer, pp. 153-188, 1998.
- [131] Zhao, Z. et al., *Visualization of Semantic Relations in Hypertext Systems*, *Proceedings of ED-MEDIA, World Conference on Educational Multimedia and Hypermedia*, Eds. Ottman, T., Tomek, I., Orlando, FL, pp. 556-564, 1993.
- [132] Ziegert, A.L., "The Role of Personality Temperament and Student learning in Principles of Economics: Further Evidence", *Journal of Economic Education*, vol. 31 (4), pp. 307-22, 2000.
- [133] Zywno, M., "A Contribution to Validation of Score Meaning for Felder-Soloman's Index of Learning Styles," *Proceedings, ASEE Annual Conference, American Society for Engineering Education*, 2003.